EXHIBIT 14

Run Length Limited (RLL)

An improvement on the MFM encoding technique used in earlier hard disks and used on all floppies is *run length limited* or *RLL*. This is a more sophisticated coding technique, or more correctly stated, "family" of techniques. I say that RLL is a family of techniques because there are two primary parameters that define how RLL works, and therefore, there are several different variations. (Of course, you don't really need to know which one your disk is using, since this is all internal to the drive anyway).

FM encoding has a simple one-to-one correspondence between the bit to be encoded and the flux reversal pattern. You only need to know the value of the current bit. MFM improves encoding efficiency over FM by more intelligently controlling where clock transitions are added into the data stream; this is enabled by considering not just the current bit but also the one before it. That's why there are is a different flux reversal pattern for a 0 preceded by another 0, and for a 0 preceded by a 1. This "looking backwards" allows improved efficiency by letting the controller consider more data in deciding when to add clock reversals.

RLL takes this technique one step further. It considers groups of several bits instead of encoding one bit at a time. The idea is to mix clock and data flux reversals to allow for even denser packing of encoded data, to improve efficiency. The two parameters that define RLL are the *run length* and the *run limit* (and hence the name). The word "run" here refers to a sequence of spaces in the output data stream without flux reversals. The run length is the *minimum* spacing between flux reversals, and the run limit is the *maximum* spacing between them. As mentioned before, the amount of time between reversals cannot be too large or the read head can get out of sync and lose track of which bit is where.

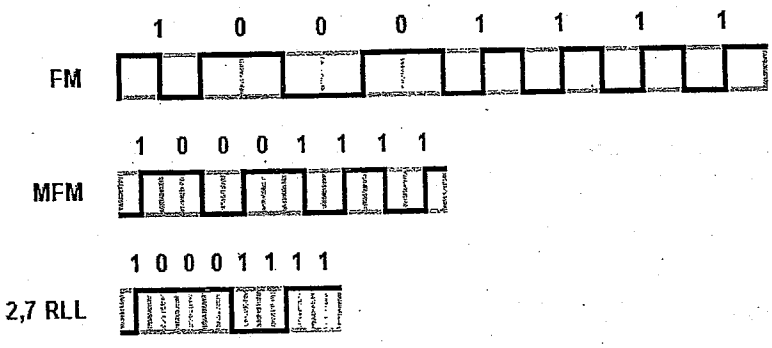
The particular variety of RLL used on a drive is expressed as "RLL (X,Y)" or "X,Y RLL" where X is the run length and Y is the run limit. The most

commonly used types of RLL in hard drives are "RLL (1,7)", also seen as "1,7 RLL"; and "RLL (2,7)" ("2,7 RLL"). Alright, now consider the spacing of potential flux reversals in the encoded magnetic stream. In the case of "2,7", this means that the the smallest number of "spaces" between flux reversals is 2, and the largest number is 7. To create this encoding, a set of patterns is used to represent various bit sequences, as shown in the table below ("R" is a reversal, "N" no reversal, just as with the other data encoding examples):

Bit Pattern	Encoding Pattern	Flux Reversals Per Bit	Bit Pattern Commonality In Random Bit Stream
11	RNNN	1/2	25%
10	NRNN	1/2	25%
011	NNRNNN	1/3	12.5%
010	RNNRNN	2/3	12.5%
000	NNNRNN	1/3	12.5%
0010	NNRNNRNN	2/4	6.25%
0011	NNNNRNNN	1/4	6.25%
Weighted Average		0.4635	100%

The controller these patterns by parsing the bit stream to be encoded, and matching the stream based on the bit patterns it encounters. If we were writing the byte "10001111" (8Fh), this would be matched as "10-0011-11" and encoded as "NRNN-NNNNRNNN-RNNN". Note that the since every pattern above ends in "NN", the minimum distance between reversals is indeed two. The maximum distance would be achieved with consecutive "0011" patterns, resulting in "NNNNRNNN-NNNNRNNN" or seven non-reversals between reversals. Thus, RLL (2,7).

Comparing the table above to the ones for FM and MFM, a few things become apparent. The most obvious is the increased complexity: seven different patterns are used, and up to four bits are considered a time for encoding. The average number of flux reversals per bit on a random bit stream pattern is 0.4635, or about 0.50. This is about a third of the requirement for FM (and about two thirds that of MFM). So relative to FM, data can be packed into one third the space. (For the example byte "10001111" we have been using, RLL requires 3 "R"s; MFM would require 7, and FM would need 13.)



2,7 RLL, FM and MFM encoding write waveform for the byte "10001111". RLL improves further on MFM by reducing the amount of space required for the same data bits to one third that required for regular FM encoding.

Due to its greater efficiency, RLL encoding has replaced MFM everywhere but on floppy disks, where MFM continues to be used for historical compatibility reasons.

Next: Partial Response, Maximum Likelihood (PRML)

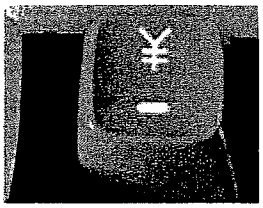
Home - Search - Topics - Up

EXHIBIT 15

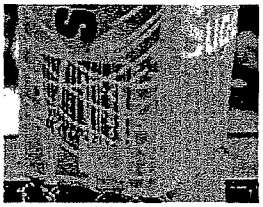
IBM

1979

Introduction **Timeline**



Since 1946, with its announcement of Chinese and Arabic ideographic character typewriters, IBM has worked to overcome **cultural and physical barriers** to the use of technology. As part of these ongoing efforts, IBM introduces in 1979 the 3270 Kanji Display Terminal; the System/34 with an ideographic feature, which processes more than 11,000 Japanese and Chinese characters; and the Audio Typing Unit for sight-impaired typists.



In the mid-1970s, IBM developed the **Universal Product Code (UPC)**, a method for embedding pricing and identification information on individual retail items. The holographic scanner technology in IBM's supermarket checkout station, introduced in 1979, is one of the first major commercial uses of holography as "wraparound" light rays read the UPC stripes on merchandise. IBM's support of the UPC concept helps lead to its widespread acceptance by retail and other industries around the world.

IBM introduces the first disk drive to feature **thin-film inductive heads** and a run-length-limited (RLL) coding scheme (IBM 3370). Thin-film heads led to a new era in higher-performance recording head design, while the "2-7" RLL code permitted higher performance while reducing errors. This leads to higher performance recording heads at reduced cost and establishes IBM's leadership in "areal density" -- storing the most data in the least space.

IBM announces the **4300** processor, featuring multilayer ceramic packaging and 64Kb memory chips that provide the densest packaging of memory and logic circuits available in intermediate-sized IBM systems; the **3279** color display terminal; and the **3287** color printer.

The first IBM retail shops, called **IBM Product Centers**, open in London and Buenos Aires.

DiscoVision Associates, a joint venture with MCA, Inc., is formed to develop, manufacture and market video discs and video disc players.

People

Employees	337,119
Stockholders	696,918

Finance

Revenue	$22.86 B + 8 %
Net earnings	$3.01 B - 3 %

EXHIBIT 16

J. M. Harker
D. W. Brede
R. E. Pattison
G. R. Santana
L. G. Taft

A Quarter Century of Disk File Innovation

This paper traces the development of disk file technology from the first disk drive to the present. A number of innovative advances are reviewed in the evolution of mechanical design, materials, and processes. These advances constitute the technological base that has permitted almost four orders of magnitude of improvement in areal density; they are discussed from four interrelated aspects: the magnetic head and its air bearing support; the head positioning actuator; the disk substrate and its magnetic coating; and the read/write signal detection and clocking electronics.

Introduction

The first issue of the *IBM Journal of Research and Development* in 1957 featured two papers, one describing the IBM 305 RAMAC system [1] and the other describing the IBM 350 disk file [2]. As Stevens mentions in his overview paper "The Evolution of Magnetic Storage" [3], the 350 was the first production movable-head disk drive. The purpose of this paper is to describe the significant innovations that led to the introduction of that product and to the evolution of the current generation of disk file products.

Inductive magnetic recording was selected as the base technology for disk files because of its advantages of nonvolatility, immediate readback without intermediate processing, unlimited reversibility, and the relative low cost and simplicity of the transducer and recording medium.

Increasing linear bit density has depended upon the scaling down of the three geometric parameters of head-to-disk spacing, read/write gap length, and disk magnetic coating thickness. Progress in reducing these key geometric parameters and the resulting linear bit density of selected products is shown in Table 1. Achieving these reduced parameters has been an iterative process and has required significant innovation. For example, scaling down the coating thickness from 1200 microinches to 25 microinches has required extensive process development in the area of substrate preparation and magnetic film coating, formulation, application, buffing, and polishing. Reducing the spacing between the head and the disk from over 800 microinches to less than 13 microinches has required an in-depth understanding of air bearing technology through analysis, simulation and testing, development of low-mass sliders with very stiff air bearings, and development of very smooth and flat disk surfaces with few mechanical asperities.

Early recording head cores were made of laminated mu-metal with the gap formed by a copper shim. Later, silicon dioxide was deposited on ferrite to form smaller, more precise gaps, and most recently the use of thin permalloy films and photolithographic techniques has permitted extremely small and accurate dimensions for head gap, pole tip width, and pole tip thickness.

Track density improvements have also been an important part of advances in disk file areal density. These improvements have come through fabrication of smaller gap widths and more accurate head positioning technology. The resulting advances in areal density are shown in Table 1.

Table 1 Development of technologies in key areas of magnetic head and its air bearing support, disk substrate and its coating, head-positioning actuator, and read/write electronics.

Year of first ship	*1957*	*1961*	*1962*	*1963*	*1966*	*1971*	*1973*	*1976*	*1979*	*1979*	*1981*
Product	350	1405	1301	1311	2314	3330	3340	3350	3310	3370	3380
Recording density											
Areal density (Mb/in.^{2})	0.002	0.009	0.026	0.051	0.22	0.78	1.69	3.07	3.8	7.8	>12
Linear bit density (bpi)	100	220	520	1025	2200	4040	5636	6425	8530	12134	15200
Track density (tpi)	20	40	50	**	100	192	300	478	450	635	>800
Key geometric parameters (microin.)											
Head-to-disk spacing	800	650	250	125	85	50	18	**	13	**	<13
Head gap length	1000	700	500	250	105	100	60	50	40	25	**
Medium thickness	1200	900	543	250	85	50	41	**	25	41	<25
Air bearing & magnetic element											
Bearing type	hydrostatic		hydrodynamic		**	**	**	**	**	**	**
Surace contour	flat	**	cylindrical		**	**	taper flat		**	**	**
Slider material	Al	**	stainless steel		ceramic	**	ferrite	**	**	ceramic	**
Core material	laminated mu-metal		**	**	ferrite	**	**	**	**	film	**
Slider/core bond	epoxy	**	**	**	**	glass	integral	**	**	deposited	**
Disk											
Diameter (in.)	24	**	**	14	**	**	**	**	8.3	14	**
Substrate thickness (in.)	0.100	**	**	0.050	**	0.075	**	**	**	**	>0.075
Rpm	1200	**	1800	1500	2400	3600	2964	3600	3125	2964	3620
Fixed/removable	fixed	**	**	removable pack		**	module	fixed	**	**	**
Data surfaces/spindle	100	**	**	10	20	19	6	15	11	12	15
Actuator											
Access geometry	x-y	**	linear radial		**	**	**	**	rotary	linear	**
Heads	2 heads/actuator		1 head/surface		**	**	2 heads/surface		1 h/s	2 h/s	**
Positioning	motor-clutch		hydraulic	**	**	voice coil motor			**	**	**
Final position	detent	**	**	**	**	servo surface			(+sector)	servo surface	
Actuators/spindle (max. no.)	3	**	2	1	**	**	**	**	1	2	**
Avg. seek time (ms)	600	**	165	150	60	30	25	**	27	20	16
Read/write electronics											
Data rate (Kbytes/s)	8.8	17.5	68	69	312	806	885	1198	1031	1859	3000
Encoding	NRZI	**	**	**	2 f	mfm	**	**	mfm	2, 7	**
Detection	ampl	**	**	**	peak	delta	**	**	**	delta clip	
Clocking	2 osc	**	clk trk	osc	vfo	**	**	**	**	**	**

**Same as in preceding column.

The evolution of the whole technology is given in the overview paper [3], and the progress in disk file manufacturing and in selected innovations in materials, processes, and testing is discussed in the paper by Mulvany and Thompson [4]. The present paper traces the development of each part of the technologies in four key areas:

- The development of the magnetic head and its air bearing support that provides the close spacing between the disk surface and read/write head necessary for high-density magnetic recording.
- The development of the disk substrate and its magnetic coating.
- The mechanical design aspects of the actuator that positions the read/write heads over concentric tracks of a rotating disk.
- The key innovations in logic and electronics required to read and write data reliably and accurately from a disk.

Much of this development has been based upon the work of many individuals who have created, over the past quarter of a century, a technological base that has permitted the improvement of almost four orders of magnitude in areal density shown in Table 1 and also enhancements in performance, function, and reliability. Two individuals in particular deserve mention because their influence was so pervasive through the early days of development. They certainly deserve credit for motivation and for active participation in many of the innovations to be discussed. They are R. B. Johnson, who had the vision that such a device was needed and could be built, and L. D. Stevens, who provided the engineering management that realized the first successful product [5].

Air bearing spacing and magnetic heads

Film bearings, both self-acting and externally pressurized, have been in common use for over a hundred years.

They allow one surface to move relative to another, with a film of either a compressible or incompressible fluid preventing contact and therefore preventing wear. A common example would be the journal bearing in an automobile engine. The use of air as the lubricating film was first reported by Kingsbury in 1897 [6]. The whole concept behind magnetic disk files rests on use of this simple mechanism. The innovation was to use such a bearing to maintain a small and consistent spacing between a magnetic read/write head and a disk surface whose axial runout was far greater than the spacing required.

• *Hydrostatic (pressurized) air bearings*

The event that provided the basis for the first disk file was accomplished on June 2, 1953, when W. A. Goddard [7] demonstrated the operation of an air-fed bearing with a magnetic element incorporated. This head, shown in Fig. 1, had small holes drilled around a circumference on its face so that air flowing out provided a pressure pad to resist the force of a spring-loaded suspension. The magnetic head mounted in the center successfully wrote and read bit patterns on a disk, establishing the feasibility of maintaining a consistent spacing between a disk surface and a read/write head. Of all developments related to the disk file, the ability to maintain a consistent spacing (and in subsequent generations, to diminish it) has been the prime driving force for all other improvements in the recording technology.

N. A. Vogel took the air bearing concept and evolved the configuration used in the first production file. It was found that a bleed hole in the center of the bearing was necessary for stability. The force required to load this head onto the disk was provided by three pins acting as pistons connected to the same air chamber that supplied the holes on the bearing surface. When air pressure was applied, the head was forced against the disk with a load proportional to the support provided by the bearing. Light springs were provided to return the head from the disk to the socket when air pressure was removed or in the event of a failure in the air supply [8, 9].

This head provided a spacing of about 800 microinches for a separate write-wide magnetic element and a read-narrow magnetic element. This combination provided a guard band between recorded tracks to allow for the positioning tolerance of the moving-head assembly.

• *Hydrodynamic (self-acting) air bearings*

While Vogel's air head design was improved through several generations of development, it became apparent that eliminating the disk-to-disk excess motion by placing one head element on each disk surface would be a significant functional improvement. Indeed, files having several air-fed heads per surface were built for the IBM Stretch computer system. The high consumption of compressed air and the maintenance and reliability associated with compressors, however, made this design of limited applicability.

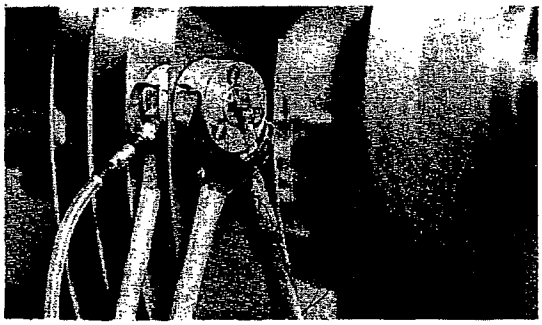
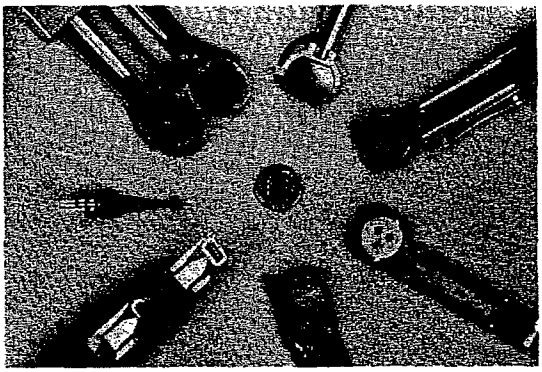


Figure 1 First air bearing magnetic head, tested on June 2, 1953.

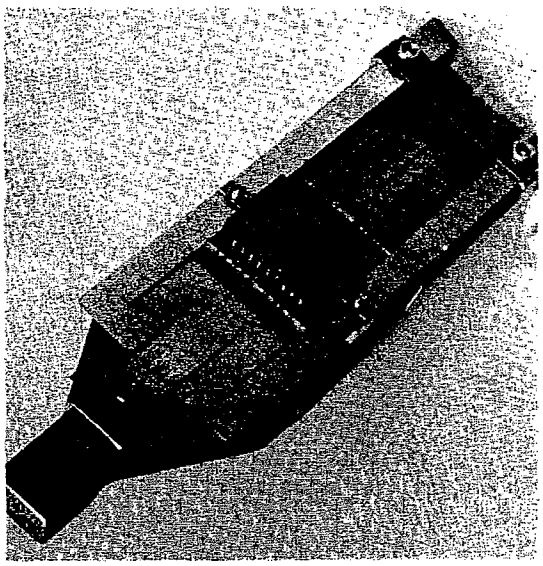
J. J. Hagopian conceived of using a self-acting air bearing in such an array of magnetic head carriers. His experimentation showed that a reasonably flat bearing properly gimbaled and loaded would generate a hydrodynamic bearing and provide stable spacing of the magnetic element from the rotating disk. Based upon this concept, an advanced disk file project was initiated. As development proceeded it became apparent that the operation of this bearing was erratic. It would fail, at times, when the slider occasionally touched the disk surface. This would often lead to catastrophic failure of the disk surface and subsequent destruction of the magnetic head in its carrier.

Classical bearing theory dealt only with an incompressible fluid. The effect of compressibility on the operation of such bearings was the subject of an intensive analytical and experimental study conducted in the summer of 1957. The analytical work was done by Dr. W. A. Gross [10] and the experimental work was done principally by Dr. K. E. Haughton and R. K. Brunner [11]. The practical result was that an entry wedge must be provided for the entry of the boundary-layer air film.

This entry wedge was obtained by providing a convex cylindrical surface on the face of the slider, which analysis and experiment indicated should have a radius of curvature of about 250 inches. An innovative way to fabricate such a slider was suggested by Brunner. First, he accurately distorted concavely the stainless steel slider



(a)



(b)

Figure 2 Evolution of the designs for slider and suspension: (a) accessing heads; (b) 2305 nine-track fixed head.

face in the direction of the boundary-layer air flow. The face was then lapped flat while it was distorted. Releasing the distorting force produced a convex cylindrical surface with a precise radius of curvature that provided the necessary entry wedge. The manufacturing process developed to produce this surface economically used the inverse of the technique described above; *i.e.*, the lap was distorted to the desired radius of curvature. Enhancements of this concept were used to produce all cylindrical slider-bearing surfaces.

These bearings would operate as predicted on flat surfaces, but there were still instability problems on the actual disk surface. Brunner and Haughton found that the local curvature of the disk surface was of the same order of magnitude as the convexity of the slider, producing regions of instability as the disk rotated. They also found that the runout of the disk produced local accelerations that caused excessive change of spacing. Specification of disk flatness in terms of the velocity and acceleration of runout and local curvature has since been basic to the disk manufacturing process [4].

Asperities on the surface of the magnetic coating, even though buffed, could cause instability of the slider. Brunner introduced the technique of running a head made of a hardened material back and forth over the disk at a spacing less than normal, thereby burnishing off any local asperities by repeated contact. This technique is still commonly used in disk manufacture.

Related to the slider-bearing technology is the design of a suspension that is simultaneously rigid in both directions in the plane of the disk but allows freedom to follow the vertical runout of the disk and provides for roll and pitch as the head follows the variations of the disk contour. It is particularly important to have a high radial suspension rigidity because of the high accelerations as the actuator moves the head and suspension from track to track.

Improvements in the self-acting slider-bearing technology, the magnetic elements, and their suspension systems have been key to increasing disk file capacity. The earlier sliders required a load of about 350 grams, as contrasted to the later generation where the required load is on the order of 20 grams. The following section discusses the development of heavy-load slider/head development and a subsequent section reviews the development of the light-load slider head.

Heavily loaded sliders, suspensions, and heads

The first disk drive to use a self-acting slider was the IBM 1301. It used a gimbal ring design that encircled the stainless steel slider and was attached to it and the supporting arm with cone-pointed pivot screws [Fig. 2(a)]. The flying height of the slider was 250 microinches and the magnetic element was composed of laminated mu-metal potted in epoxy in a small can that provided magnetic shielding. The can was in turn epoxied into the slider. A linear density of 520 bits per inch (bpi) and a data rate of 544 Kbits/s was achieved with this combination. The gimbal ring provided freedom for roll and pitch and stiffness for yaw, resisting both radial and circumferential forces. It carried the 350-gram load from a long torsion spring pressing on the arm through the gimbal to the slider. This design was difficult to maintain in that the

pivot screws required adjustment and lubrication. Adjustment was disturbed by temperature variations, and oil sometimes found its way to the disk surface.

This basic suspension design was improved in several steps, as seen in Fig. 2(a). The most significant improvement was made with the design for the IBM 2311. It separated the vertical load requirements from the gimbal system. The load was applied directly to the slider at the center of roll and pitch. The 2311 suspension design used a thin flat flexure ring to control the roll, pitch, and yaw. The flexure ring was cut from stainless sheet stock and the joints were made by spot welding, eliminating screws and lubrication and reducing material, fabrication, and assembly cost. Variants of this design were used on all subsequent sliders.

The increased track density of the 2311 [from 50 to 100 tracks per inch (tpi)] was achieved in part by use of an improved head element configuration. This involved a change to a "tunnel" erase trimming of the newly written track [12] that provided a better signal-to-noise ratio than the previous guard band techniques of write-wide read-narrow and erase-wide write/read-narrow. Also introduced on the 2311 head was a ball staking technique for mounting the magnetic element in the slider [13]. This technique reduced element movement within the slider assembly and maintained the tighter tolerances necessary for the higher track density.

A ferrite magnetic head element was designed for the IBM 2314 disk file, first shipped in 1966, because of its 2.5-megabit/s data rate resulting from increased bit density (1100 to 2200 bpi). Manufacturing process considerations dictated that these ferrite magnetic elements be mounted in a mechanically similar ceramic material. Alumina was selected because it could be formed to the desired circular geometry and had desirable mechanical properties. It provided an improved head/disk interface, and ceramic or ferrite materials have been used in all subsequent products. The ferrite element was bonded to the slider with an epoxy and the face of the assembly lapped to the required cylindrical surface.

As the head-to-disk spacing was reduced from 85 microinches for the 2314 to 50 microinches for the IBM 3330, it became critical that the magnetic element gap be flush with the face of the slider. In addition, the thermal and hydroscopic properties of epoxies used to bond the ferrite core to the slider became inadequate. Consequently, a compatible ferrite-ceramic-glass material combination was developed for the 3330. This required two compatible glasses with enough separation in working temperatures to allow the separate assembly bonding of the parts of the core and the parts of the slider with a high-temperature glass. Then the final assembly of the previously assembled core and slider was accomplished with a second glass at a lower temperature that did not disturb the dimensions established in the first bonding [14]. A new slider ceramic was developed that had a thermal expansion coefficient compatible with ferrite and glass, and was of sufficient hardness, density, and grain structure to provide a satisfactory bearing surface.

The 3330 slider geometry was changed to rectangular and the bleed holes were replaced by a central slot [Fig. 2(a)]. A spring element in each support arm provided the mechanical load for the associated slider [15]. This design reduced the tolerances associated with the mechanical load and made it possible to set the flying height of each slider to close limits. This combination of materials and geometry produced a dimensionally stable head/slider assembly and made possible a head-to-disk spacing of 50 microinches for the 3330-1, first shipped in 1971, and 35 microinches for the 3330-11, first shipped in 1974. Both products had a bit density of 4040 bpi. The track density of the Model 1 was 192 tpi and that of the Model 11 was 370 tpi.

The head/slider combination developed for the IBM 2305 fixed-head file, a product complementary to the 3330 and also first shipped in 1971, was based upon the concept of a batch manufacturing process developed by E. R. Solyst [16]. In this design, nine head elements and a flat slider with a taper at the leading edge were machined from a single block of ferrite [Fig. 2(b)]. This taper-flat design, with its large air bearing surface, required a loading force of about 1.2 kilograms to achieve a 50-microinch flying height.

Low-mass, lightly loaded sliders, suspensions, and heads

Winchester taper-flat slider The next major step in slider and head development came with the introduction of the IBM 3340, first shipped in 1973. This head combined the taper-flat and batch-fabrication concepts developed for the 2305 with the experience gained at the IBM Los Gatos Laboratory, where a unique file had been developed in the late 1960s using disk technology to provide a refresh buffer for a high-resolution display system. This fixed-head file, developed by Dr. J. T. Ma, was the first to use lubricated iron oxide particulate disks and start-and-stop-in-contact heads. To achieve the linear density required for that buffer, heads developed by Data Disk Corporation for use on a plated disk were used. These heads had three small pads arranged in a triangle with the magnetic element at the apex providing one of

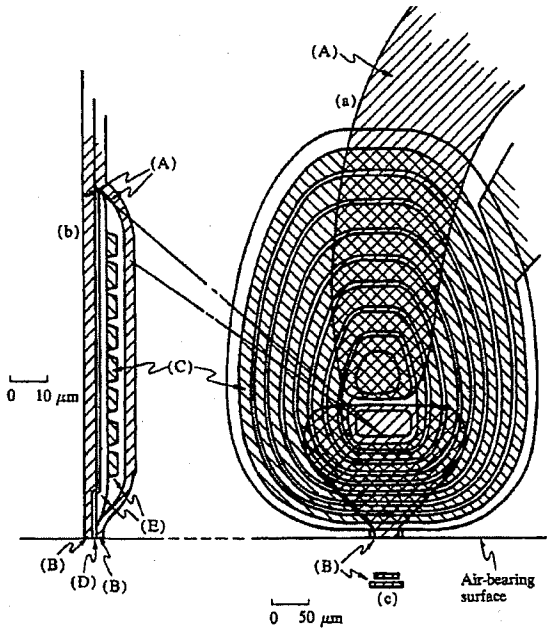


Figure 3 IBM 3370 thin-film head: (a) planar view; (b) center line cross section showing A—magnetic layers, B—pole tips, C—conductor turns, D—gap layer, E—insulation layer; (c) pole tip geometry at the air bearing surface (from Ref. 21).

the pads. They provided a small air bearing area and a low mass, were lightly loaded with about 20 grams by a flexure suspension, and were in contact with the disk surface when it was stopped. In operation they maintained a nominal spacing of 20 microinches. A lubricated particulate disk was developed by IBM to minimize wear during the periods of in-contact operation. IBM obtained rights to produce this tri-pad head design and a new version was developed by W. S. Buslik for use in a disk drive embedded in the IBM 3735 Programmable Buffered Terminal.

Experience had shown that this tri-pad design was difficult and costly to manufacture, and in 1969 a development effort was initiated to design a head that would operate at close spacing but could be produced in quantity with a simple process. The resulting tri-rail design by Warner [17] is known as the "Winchester" head. A small taper-flat bearing area, such as in the 2305, was provided by the outer two rails of the tri-rail design. The center rail defined the width of the magnetic element at the trailing edge where a ferrite core was formed, as in the 2305 design. This head maintained a spacing of about 20 microinches with a load of less than 20 grams. Mulvany [18] gives a more detailed discussion of the evolution of the Winchester head/slider for the interested reader. The elegant simplicity of this design has given it industry-wide acceptance.

Film magnetic elements Based upon the prior development work of plated magnetic memories, work was begun on film recording heads in the late 1960s. Romankiw, Croll, and Hatzakis [19] built the first operating thin-film head in IBM. The success of these early designs resulted in a development activity to integrate a film magnetic element with a slider bearing in a configuration suitable for product use. Additional contributions by Thompson and Romankiw [20], Jones [21], and others led to the design of the head that was first shipped in the IBM 3370 in 1979. The advantages of this structure, shown in Fig. 3, are the precise control of the gap, the inherently thin pole tips which enhance the sharpness of the readback pulse, and the excellent frequency response of the element. These combine to give a unique potential for enhancement of disk file capacity and data rate.

Design aspects of disk file actuators

The key mechanical components of a moving-head disk file are in the actuator and carriage assembly [22]. Its role is to transport the read/write heads from one track position to another as fast and as accurately as possible. Speed is important because it directly affects seek time, which is a major component of access time. Positioning accuracy is important because it affects the maximum attainable track density. In this section, the progress of actuator and carriage development will be reviewed with emphasis on the improvements made in both seek time and positioning accuracy.

- *The IBM 350 actuator*

The IBM 350 disk file [2] provides a reference to measure progress. A schematic of the actuator and carriage of the 350 is shown in Fig. 4. Two pressurized air-bearing-supported magnetic heads, gimbaled to a pair of arms, were mounted on a carriage that was moved vertically to one of 50 disks 24 inches in diameter and spaced 0.4 inch apart, by a cable connected through two counter-rotating magnetic powder clutches driven by a single motor. A null servo system and then a mechanical detent were used to position the carriage at the selected disk. Detenting the carriage changed the mode of the actuator, permitting the same drive system and a similar null servo to move the two head-arm assemblies to one of 100 concentric tracks in a five-inch radial band. Upon arrival at the selected track, air pressure was applied to detent the arm into its final position and load the air bearing heads against the disk surface preparatory to reading or writing. The detenting system provided a track density of 20 tpi. The average seek time for this two-dimensional actuator was 600 milliseconds.

Hydraulic actuators and comb access structures

The next 24-inch-diameter disk file, the IBM 1301, introduced the self-acting air bearing slider in place of the externally pressurized bearing. This made the development of a comb-type carriage practical, with one self-acting slider for each disk surface. A new hydraulic actuator was developed to increase the access rate, and the comb of heads provided a logical cylinder of data tracks at each access position. This actuator employed two sets of hydraulic cylinders, one each for coarse and fine positioning, and was capable of moving the carriage with its 50 head-arm assemblies from one logical data cylinder to any other with an average seek time of 165 milliseconds. The accuracy of the hydraulic system with detents permitted an initial track density of 50 tpi, later increased to 100 tpi.

Removable disk pack

When it was seen that needs of small systems users could not be met at an acceptable cost using the large components described, a proposal was made for a small disk file, utilizing a removable disk assembly to allow off-line storage and backup [23]. Consequently, a disk drive was developed [24], using six 14-inch-diameter disks mounted in a customer-interchangeable disk pack enclosed in a protective cover that could be removed and replaced only when the pack was in place on the drive hub [25, 26]. The IBM 1311 was the first of a family of files with interchangeable disk packs. This design concept was extended into the early 1970s.

The design of an interchangeable disk pack required tight control of the absolute position and the position tolerances of the data tracks written on each pack. Control of the absolute position of the tracks was established by introducing a special disk pack for use by both manufacturing and field engineering. A cylinder of tracks was very precisely written on this pack using bit patterns chosen for ease in manually adjusting data heads. This special pack permitted all of the heads on all of the drives to be initially aligned to the same absolute cylinder location. The location of all of the other cylinders on the disk pack was then a function of the carriage, disk pack, temperature, and spindle tolerances of the disk file that wrote data on these cylinders.

Table 2 lists the key tolerances affecting track positioning accuracy for several file systems with interchangeable magnetic media. The track positioning accuracy can be determined by combining these tolerances statistically. The result is referred to as the "track misregistration tolerance." Track density in all disk files is determined by the condition that the head, disk, and recording channel off-track capability be greater than or equal to the track misregistration tolerance. As already noted, the early disk drives using interchangeable disk packs were able to achieve track densities of 50 tpi.

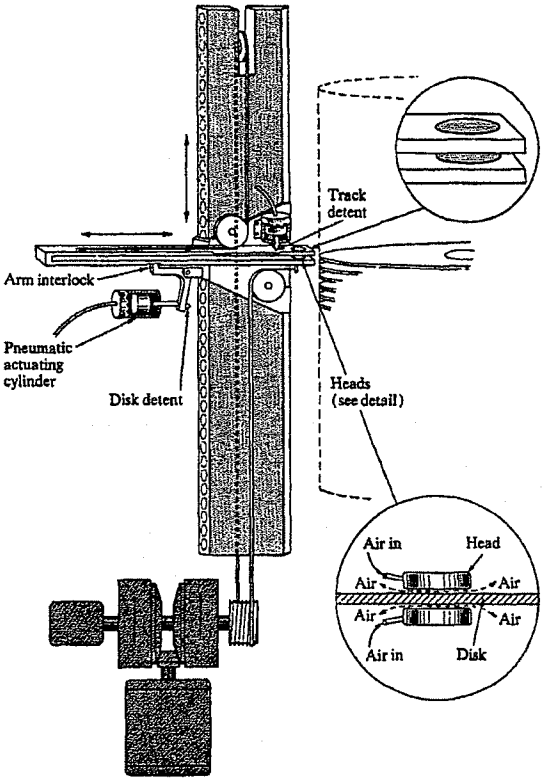


Figure 4 IBM 350 carriage and actuator (from Ref. 2).

Table 2 Position tolerances* affecting track positioning accuracy (microinches).

Item	*1311*	*3330*	*3340*
Spindle eccentricity	250	**	**
Spindle/pack tilt	250	100	15
Carriage—radial position	500	190	246
Carriage—circumferential	310	240	
Alignment cylinder	200	100	0
Head adjustment	200	80	0
Vibration	150	100	103
Temperature	500	100	235
Wear	250	50	25
Servo		400	410

*These tolerances are "single-sided"; *i.e.*, a tolerance of 250 means a total error of ±250.
**Included under servo.

Disk drives with small removable disk packs required a simple low-cost high-performance actuator and carriage

AXD024632

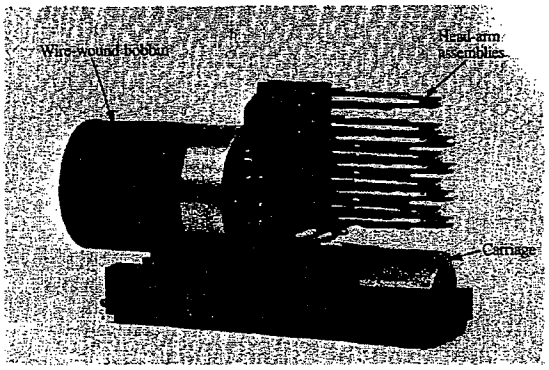


Figure 5 IBM 3330 voice coil actuator.

assembly. An innovative hydraulic actuator was developed that took advantage of the shorter stroke length and reduced carriage mass. The IBM 1311 actuator drove the carriage selectively at one of two speeds determined by fluid flow rates. A high flow rate (and high speed) was used for traveling long distances, and a low flow rate (and low speed) was selected just prior to arriving at the selected data cylinder. At the low speed, a mechanical detent on the carriage could be engaged without creating excessive carriage vibration and overshoot. For short distances, only the low speed was employed. Hydraulic fluid was delivered to this two-speed actuator by a small commercially available gear pump. This combination resulted in a low-cost carriage and actuator assembly with an average seek time of 150 milliseconds.

The basic actuator and carriage concept of the 1311 was also employed on successor products, the 2311 Disk Drive and the 2314 File Facility. Carriage tolerance modifications achieved the more stringent track density requirements of 100 tpi. With the change from a two-speed to a three-speed actuator the average seek time was reduced from 150 milliseconds to 60 milliseconds.

- *Track-following servo actuators*

The 1311 tolerances listed in Table 2 show that the radial positioning accuracy of the carriage and the temperature effects contribute almost 40% of the total. This was so because the radial positioning detent was referenced to the main drive casting rather than a track on the disk pack. There was no compensation for even a portion of the nonuniform and time-varying effects of temperature. By moving the reference from the base casting to the disk pack and providing a feedback loop to control the carriage position, a portion of the tolerance due to temperature could be eliminated.

The IBM 3330 was the first production disk file to incorporate a track-following feedback control system. Feasibility of a similar positioning concept was demonstrated by Hoagland in the early 1960s [27]. The 3330 feedback system consisted of

- One dedicated disk surface with special prerecorded tracks defining the position of each data cylinder.
- A servo head to read the prerecorded tracks.
- An electronic control system that generated a position error signal and translated it into actuator drive current.
- An actuator motor that generated force directly proportional to drive current.

The actuator was made of a wire-wound cylindrical bobbin supported by a carriage (Fig. 5) and moving in a magnetic field structure similar to a loudspeaker (hence the term "voice coil motor"). During seeking, current in the bobbin was controlled by monitoring servo track crossings and resulted in heads arriving at the selected cylinder with very little overshoot. Feedback control was continuously provided in the track following mode, permitting the servo head to follow the radial runout and temperature effects. The signal pattern recorded on the servo tracks [28] was a key factor in attaining accurate track following at a reasonable cost.

Tolerance improvements in other areas are shown in Table 2. The net effect of all tolerance improvements resulted in attaining 192 tpi. The servo control system with the voice coil motor resulted in an average seek time of 30 milliseconds.

With the development of the 3330, the 1311 family had grown from a single spindle disk file of 2.68 million bytes to a dual spindle file of 100 million bytes per spindle. No longer was this file family designed for the first-time file user but rather for the experienced disk file system user, leaving a need for a smaller-capacity and lower-cost entry-level disk file. To meet this need, the IBM 3340 disk file was developed.

The IBM 3340 actuator and data module

The IBM 3340 contained many innovative designs [18]. Key was the low-mass, lightly loaded "Winchester" head and lubricated disks which permitted heads to remain on the disk while it was starting and stopping, thus eliminating the load/unload mechanism. Because of the low cost of the head/slider, two heads per surface could be used. This cut in half the stroke length and lowered the cost of both the carriage and the actuator. The 70-megabyte capacity was obtained with four disks. The disks, the disk spindle and bearings, the carriage, and the head-arm assemblies were incorporated into one removable package called a Data Module [18]. Since each head need only

read the data that it had written—a concept that had been suggested earlier by Buslik [29]—the need for a special disk pack and for adjusting the individual data heads was eliminated. The positioning accuracy tolerances are also shown in Table 2. The IBM 3348 Data Module attained a track density of 300 tpi and an access time of 25 milliseconds.

The IBM 3350, first shipped in 1976, extended the Winchester technology by increasing the number of disks per drive, the bit density to 6425 bpi, the track density to 478 tpi, and the data rate to 1.2 Mbytes/s, resulting in a 4.5 times increase in capacity per spindle to 317.5 Mbytes. One of the key changes that permitted these improvements was the decision not to provide a customer-removable data module. Tolerances associated with removability were eliminated, allowing an increase in track density. Eliminating removability reduced still further the potential for particulate contamination due to handling of the data module.

Rotary actuators

An ingenious actuator [30] developed at the IBM Development Laboratory in Hursley, England, involved a moving-coil rotary actuator, Fig. 6, with pivoting data arms. Although the pivoting data arm concept was not new, this rotary actuator and carriage, with only one moving part, provided most of the advantages of the linear voice coil actuator [31], but at very low product cost. As a result this actuator using the Winchester head technology, combined with a nonremovable disk in a sealed enclosure, has been used in imbedded disk files for several small systems such as the IBM System/32. The heat generated by this small unit was low enough to be dissipated directly through the enclosure, without requiring external air cooling, thus providing low-cost protection from external contamination.

The latest small IBM disk drive developed at the Hursley Laboratory, the IBM 3310, also utilizes a rotary actuator and carriage with a dedicated servo surface. In addition, it employs a sectorized track-following servo concept [32, 33]. Each data surface has prerecorded feedback information located in small radial sectors equally distributed around the disk. During seek operations the dedicated servo surface is used to provide feedback information, but when the desired cylinder is reached, position information from the selected data head is combined with the information from the dedicated servo surface to control the carriage position. The control electronics switches data and servo information to appropriate circuits and samples and holds servo information between sectors. This sector servo control virtually eliminates temperature effects and most of the tolerances between the dedicated servo head and the data head.

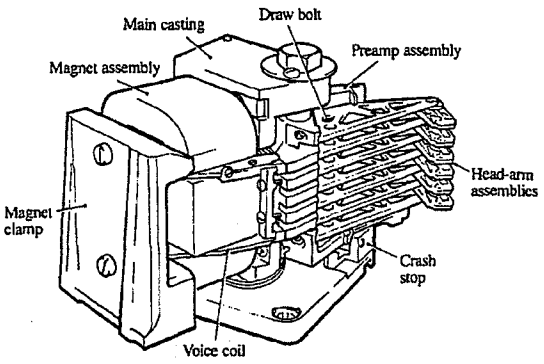


Figure 6 Rotary actuator used in the IBM 62 PC.

Disk substrates and magnetic coating

Substrates In early disk file development the major concern with the substrate was its axial runout. The first disk file used an aluminum substrate of "recording stock" quality and disks were fabricated by laminating two half-disks, 0.051 inch thick, together with an adhesive. The coating on these disks was relatively thick, 1200 microinches, masking the effects of substrate imperfections.

The requirements for higher-density recording demand that the head be closer to the surface and the magnetic coating thickness be reduced. As the thickness of the magnetic coating and the flying height of the head/slider decreased, the effects of the substrate became more important. Various polishing, grinding, and machining techniques were developed to improve flatness and smoothness. A key advance in the fabrication of substrates was the development of a diamond tool-lathe process in conjunction with the use of a much purer aluminum alloy [4]. This process, developed by the Manufacturing Group in Sindelfingen, Germany, established a technology base that has provided substrates for all subsequent generations of IBM disk products.

New and better measurement techniques were developed. The measurements of the first and second derivatives of runout were important in predicting the excursion of the head over local surface imperfections. With the introduction of lower flying heights and increasing rotational speed of the disk, the resonant frequencies of the substrate became important. Rotational speed and substrate thickness were selected to avoid these frequencies.

Magnetic coating The magnetic coating is a dispersion of magnetic particles in a binder. In the early 1950s the magnetic particles were limited to the gamma form of iron

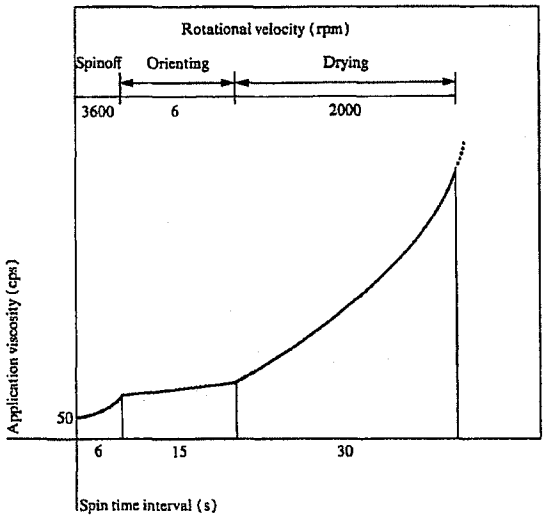


Figure 7 Plot of time required to orient magnetic particles during coating a disk, *vs* the fluid viscosity of the magnetic medium.

oxide, and polymer chemistry was in its infancy, providing few choices of polymeric materials to satisfy the rigorous demands for a disk recording medium. The key demands on a binder material are

- Ready dispersion of magnetic iron oxide particles.
- Viscosity low enough to spin-coat on an aluminum substrate.
- Curing to form a crosslinked, hard, durable material.
- Good adhesion to the substrate.
- Smooth surface polish with few pinholes or other imperfections.

Literally thousands of combinations were tried through much empirical study and diligent experimentation. The work culminated in a unique binder system described by Johnson, Flores, and Vogel [34]. The choice of a combination of an epoxy, a phenolic, and a polyvinyl methyl ether proved to be an excellent selection and with modification is still the basic formulation in use today [4]. Similarly, the original spin coating used for the first disks continues to be the predominate process for most disk production today.

The magnetic properties of commercially available magnetic particles have continued to improve over the years. Good control in particle geometry yields materials that facilitate particle orientation within the recording medium. Structural defects within particles have also been improved by minimizing pores and nonmagnetic inclusions between the individual crystallites of a particle, thus leading to better control over the range of particle-switching fields.

Magnetic particle orientation Orientation of the oxide particles as routinely practiced in magnetic tape had long been an objective of disk development. The geometry of a disk and the requirements of performing the orientation on the spin coater proved a difficult hurdle for many years. E. M. Williams found that a new iron oxide particle from Pfizer had high acicularity and a narrow switching field distribution. This oxide and the concurrent work on orienting particles on disks by P. T. Chang, A. W. Ward, and W. N. Johnson led to a new disk product with substantial improvement in recording properties. The oriented disk was first introduced in the 3340 Data Module and has been used in all subsequent disk file products.

The key parameters for orientation are the fluid viscosity of the media, the time required to orient the particles, and the magnetic field applied to the particles. It is necessary to carefully control both the spinoff conditions and duration as well as the application viscosity of the fluid, as indicated in Fig. 7. Because the orienting field is from the gap fringe field of a magnet, both the strength of the field and the time the particles "see" that field become the critical factors for good orientation. With a pair of opposing magnets on each side of the disk so that the fields are essentially in the plane of the disk, it was found that a gap field of 1800–2000 oersteds [1 Oe = $(1000/4\pi)$ A-m^{-1}] was required. If the field was weaker than that, there was insufficient force to rotate the particles in the viscous medium. If the field was much bigger than that, the particles simply switched their magnetization polarity rather than being rotated. It was also necessary to rotate the disk slowly for maximum orientation. If the disk were rotated too rapidly, the particles switched their magnetization polarity, and if too slowly, it caused coating imperfections.

Lubrication The development of the low-load Winchester slider, designed to be in contact with the disk surface during starting and stopping, put new requirements on the surface characteristics of the magnetic medium. These requirements were met by the development of lubricating techniques for disks similar to those previously used on magnetic tape. The first lubricant used a 0.75% solution of Dow Corning RDC-200 in a freon solvent, and was applied topographically to the disk with a soft absorbent cloth. This disk proved to be very reliable and led to the development of disk file products utilizing the concept of a lubricated surface with start-and-stop-in-contact heads.

In succeeding years, numerous other lubricants were evaluated to meet the demanding needs of the newer high-performance files.

Data encoding and channel electronics

Advances in encoding methods and recording electronics have also contributed to increasing disk file areal densities. As linear bit densities have increased from 100 bpi to 15 000 bpi and rotational velocity from 1200 to 3600 rpm, the serial data rate at the read/write head has increased from 100 kilobits/s to 24 megabits/s. These data rates have, in recent generations, pressed the data rate capability of host system data channels, and in addition have provided a challenge in the design of disk file electronics.

For writing data, the electronics converts logic-level data signals into a current that drives the recording head. Later, when data are being read, the process must be reversed. Analog signals induced in the coils of the read head must be amplified, detected, and clocked. Reliable recovery of stored information involves both time detection and amplitude detection. Encoding methods have been designed to place more emphasis on either one or the other of these two dimensions of detection, depending upon the disk and head magnetic and electrical characteristics.

The first disk file utilized NRZI encoding [35], developed initially for tape drives. Data were recorded by changing the direction of current flow in the head for every 1-bit, with no change in current direction for a 0-bit. Sufficient current was applied to saturate the medium. Since the readback signal is proportional to the rate of change of recorded field, a pulse was read back for every 1-bit. Presence of the signal was detected when the head voltage exceeded a predetermined clipping level. This amplitude detection was capable of discriminating against background noise, base-line overshoot, and extra bit defects. The data-clocking circuits, developed by Seader [36], used two fast-starting oscillators, each started by alternate 1-bits. After starting, the cycles of one oscillator were used to clock any following 0-bits. The second oscillator was started when the next 1-bit was detected. The use of an odd redundancy bit and a space bit at the end of each character limited the free-running cycles of an oscillator to at most one character.

The next disk file used a combination of techniques for developing a clock. A separate clock track provided a stream of periodic pulses with the rotation of the disk. Each bit time was divided into four phases by delaying the base clock three times. At the beginning of each byte or character of data, one of the four phases was selected to clock the remaining data bits. Since the clock track signal was locked to minor disk speed variations, and with the time reference re-established at the beginning of each byte, the expected time variation between a clock phase and data from a head was small. Furthermore, mechanical variations due to vibration and temperature differences between the clock head and data head, which could cause clocking errors, were low in frequency relative to the data stream. The major drawback of this system was the cost of a clock head and read amplifier together with logic to perform the phase selection. The amplitude detection system was improved by the addition of automatic gain control to eliminate the wide range of signal amplitude variation caused by different head and disk configurations. As bit densities increased, so did data rates, and it became more important to generate a clock that was synchronized with the data to provide additional clocking tolerance.

The variable-frequency oscillator (VFO) was a circuit which offered the potential of reducing many tolerances experienced by previously described circuits. The VFO had been developed by Newman [37] for tape but was redesigned for the high data rate of disk files. The circuit was an oscillator whose frequency could be electronically adjusted with a feedback loop. Recorded pulse rates were compared with an internal oscillator. If the oscillator was operating at too low a frequency, an error signal was generated to increase the oscillator frequency. The circuit eventually locked onto the periodicity of data pulses, but lagged a slight amount to generate a continuous error signal. An improvement of this circuit by Lang, LaPine, and Vaughn [38] integrated the error signal and drove the phase as well as the frequency error to zero. The averaging properties of this circuit reduced to almost zero those errors caused by setting the clocking window. The primary tolerance of clocking was now associated with the data bit rather than the location of the clocking window. Initially, double-frequency encoding was used, providing a clock pulse with every bit cell as a point of reference.

The clocking accuracy of a VFO operating at high frequencies permitted utilization of an encoding method with reduced transitions but required two clock cycles per bit cell. The structure of the code made it slightly more susceptible to noise, and a new detection system known as the delta-V detector was developed by Graham, Scovmand, and Swartz [39] to maintain and improve reliability. The delta-V detector used a series of tests to improve the signal rejection capability over simple amplitude discrimination. A minimum leading edge slope was required, and the signal peak was detected by finding the derivative crossover point, and a minimum trailing slope was sensed. Finally, the amplitude of the signal associated with the pulse had to exceed a minimum level.

Through the use of a delay line and logic, these four conditions were tested, and if they were satisfied, an output pulse was generated.

More recently, Franaszek-type run-length-limited codes [40] have been implemented by Eggenberger and Hodges [41]. These codes limit the minimum and maximum distance between transitions. For example, the 2,7 code has a minimum of two zeros and a maximum of seven zeros between transitions. At the same time they use the transitions with such efficiency that on the average 1.5 bits per transition is achieved. A new method of detection was developed by D. B. Chapman for use on recent IBM disk products with such a 2,7 run-length-limited code. The signal is tested to see that the polarity of the pulses alternates and that a minimum amplitude is achieved. The signal peak is found through differentiation, and finally, a minimum signal change, occurring after the peak, must occur within a specific period of time. These four criteria are logically brought together to make a detection decision.

As track densities increased, the readback signal amplitude was reduced. Furthermore, increasing data rates led to introduction of an electronics module on each access arm. This module provided the write current as well as amplification for the read signal. Since every arm assembly was designed to use one module containing a total of 150 high-speed bipolar circuits, low cost was important to make this system practical.

Summary

When the first disk file was conceived in IBM's newly established development laboratory in California, no one visualized how far the technology would be extended. The first disk file introduced the externally pressurized air bearing as a support for the magnetic head, the 24-inch disk, the coating technology, and a two-dimensional access mechanism. The starting point of areal density was 100 bpi linearly and 20 tpi radially, or 2000 bits per square inch.

The next generation introduced the use of self-acting bearings as a head carrier and the high-performance comb actuator. Areal density was increased to 100 000 bits per square inch (100 tpi and 1000 bpi), a factor of 50 improvement. This file, part of the IBM Sabre system for airline reservations, was key to the start of interactive processing as it is known today.

The disk pack files extended usage to low-end systems. It made the disk drive the primary storage for such systems. The interchangeable pack permitted it to serve both tape and disk functions. A number of the small systems were supported by disks exclusively.

Further increases in capacity resulted from continuing work in mechanics, electronics, materials, and processes, which further improved storage density. This continuing evolution lowered the cost of on-line disk storage to a point where it became the work storage for all data processing systems, large and small alike, and permitted the expansion of on-line applications in the 1960s and early 1970s. In addition to substantial improvements in heads, disks, and recording electronics, the introduction of the track-following servo improved both performance and track density significantly.

Introduction of the Data Module drive with heads in contact with the disk as it was started and stopped provided a basis for yet another step increase of density.

With the introduction of the IBM 3350, a full circle of disk file concepts had been made. Nonremovable data disks had first been introduced in the IBM 350 design. Next came the "unlimited capacity" family with customer-removable disk packs or data modules. As the storage technology improved over the years, a removable disk file product which held two million bytes in 1963 could, in 1974, hold 200 million bytes, greatly reducing the customer needs for off-line storage. With the significant increases in reliability and availability made during the same period, users no longer had a need for disk interchangeability to meet data availability requirements. Therefore, the IBM 3350, with over 300 million bytes per spindle, was the first file of the 1970s to have nonremovable disks. That this way of reducing tolerances led to further increases in density is witnessed by the fact that the most recently announced disk file product, the IBM 3380, shipped in 1981, is also a fixed-disk product with 1250 million bytes of storage per spindle—a vast improvement over the five million characters first introduced a quarter century earlier.

References

1. M. L. Lesser and J. W. Haanstra, "The Random-Access Memory Accounting Machine—I. System Organization of the IBM 305," *IBM J. Res. Develop.* **1**, 62–71 (1957).
2. T. Noyes and W. E. Dickinson, "The Random-Access Memory Accounting Machine—II. The Magnetic-Disk, Random-Access Memory," *IBM J. Res. Develop.* **1**, 72–75 (1957).
3. L. D. Stevens, "The Evolution of Magnetic Storage," *IBM J. Res. Develop.* **25**, 663–675 (1981, this issue).
4. R. B. Mulvany and L. H. Thompson, "Innovations in Disk File Manufacturing," *IBM J. Res. Develop.* **25**, 711–723 (1981, this issue).
5. M. E. Wolf, "The R & D Bootleggers: Inventing Against the Odds," *IEEE Spectrum* **12**, 38–45 (1975).
6. A. Kingsbury, "Experiments With an Air-Lubricated Journal," *J. Amer. Soc. Naval Engineers* **9**, 267–292 (1897).

7. W. A. Goddard, "Air Bearing Head," U.S. Patent 3,213,461, 1965.
8. N. A. Vogel, "Air Head," U.S. Patent 2,886,651, 1955.
9. T. G. Cowan, "Mounting and Operating Means for an Air Head," U.S. Patent 3,018,152, 1958.
10. W. A. Gross, "A Gas Film Lubrication Study, Part I: Some Theoretical Analyses of Slider Bearings," *IBM J. Res. Develop.* **3**, 237–255 (1959).
11. R. K. Brunner, J. M. Harker, K. E. Haughton, and A. G. Osterlund, "A Gas Film Lubrication Study, Part III: Experimental Investigation of Pivoted Slider Bearings," *IBM J. Res. Develop.* **3**, 260–274 (1959).
12. W. E. Tibbetts, "Tunnel Erase Magnetic Transducer," U.S. Patent 3,155,949, 1961.
13. E. R. Solyst, "Mounting of a Transducer," U.S. Patent 3,329,379, 1967.
14. D. W. Brede, M. H. Cook, M. E. Freeman, and H. L. Turk, "Glass Bonded Head Assembly," U.S. Patent 3,610,837, 1971.
15. R. A. Applequist, "Magnetic Head Spring Bias," U.S. Patent 3,579,213, 1971.
16. E. R. Solyst, "Multichannel Head," U.S. Patent 3,579,214, 1971.
17. M. W. Warner, "Flying Magnetic Transducer Assembly Having Three Rails," U.S. Patent 3,823,416, 1974.
18. R. B. Mulvany, "Engineering Design of a Disk Storage Facility with Data Modules" *IBM J. Res. Develop.* **18**, 489–505 (1974).
19. L. T. Romankiw, I. M. Croll, and M. Hatzakis, "Batch Fabricated Thin-Film Magnetic Recording Heads," *IEEE Trans. Magnetics* **MAG-6**, 597–601 (1970).
20. David A. Thompson and Lubomyr T. Romankiw, "Film Head Development," *Disk Storage Technology*, 3–5 (1980); Order No. GA26-1665-0; available through IBM branch offices.
21. Robert E. Jones, Jr., "IBM 3370 Film Head Design and Fabrication," *op. cit.* Ref. 20, pp. 6–9.
22. W. A. Goddard and J. J. Lynott, "Direct Access Storage Device," U.S. Patent 3,503,060, 1970.
23. J. M. Harker, "Objectives for a Low Cost Random Access File Program," Internal IBM report, February 9, 1959.
24. J. D. Carothers, R. K. Brunner, J. L. Dawson, M. O. Halfhill, and R. E. Kubec, "A New High Density Recording System: The IBM 1311 Disk Storage Drive with Interchangeable Disk Packs," *AFIPS Conference Proceedings* 24 (1963 Fall Joint Computer Conference, Las Vegas), 327–340 (1963).
25. R. E. Pattison, "Portable Memory for Data Processing Machine," U.S. Patent 3,176,281, 1965.
26. T. G. Leary, "Transporting and Protecting Cases for Drum and Disk Records," U.S. Patent 3,206,214, 1965.
27. A. S. Hoagland, "A High Track-Density Servo-Access System for Magnetic Recording Disk Storage," *IBM J. Res. Develop.* **5**, 287–296 (1961).
28. G. R. Santana, "Method and Apparatus for Recording and Detecting Information," U.S. Patent 3,534,344, 1970.
29. W. S. Buslik, "Magnetic Disk Storage File in Sealed Enclosure," U.S. Patent 3,710,357, 1973.
30. D. E. Cuzner, C. O. R. Dodman, J. S. Heath, and L. J. Rigby, "Magnetic Disk Apparatus," U.S. Patent 3,849,800, 1974.
31. J. S. Heath, "Design of a Swinging Arm Actuator for a Disk File," *IBM J. Res. Develop.* **20**, 389–397 (1976).
32. R. J. Sippel, "Transducer Positioning System Utilizing Record With Interspersed Data and Positioning Information," U.S. Patent 3,185,972, 1965.
33. Robert D. Commander and J. R. Taylor, "Servo Design for an Eight-Inch Disk File," *op. cit.* Ref. 20, pp. 90–98.
34. D. D. Johnson, R. Flores, and M. J. Vogel, "Composition of Epoxide Resin, Methylol Phenol Ether, Polyvinyl Methyl Ether, and Acid Anhydride Catalyst, and Metal Substrate Therewith, Especially a Magnetic Signal Storage Device," U.S. Patent 3,058,844, 1962.
35. B. E. Phelps, "Magnetic Recording Method," U.S. Patent 2,774,646, 1956.
36. L. D. Seader, "A Self-Clocking System for Information Transfer," *IBM J. Res. Develop.* **1**, 181–184 (1957); also U.S. Patent 2,864,078, 1958.
37. E. G. Newman, "Magnetic Recording System," U.S. Patent 3,197,739, 1975.
38. P. C. Lang, A. N. LaPine, and J. E. Vaughn, "Random Binary Data Signal Frequency and Phase Compensation Circuit," U.S. Patent 3,701,039, 1972.
39. I. H. Graham, H. A. Skovmand, and J. S. Swartz, "Signal Detection Circuit," U.S. Patent 3,631,263, 1971.
40. P. A. Franaszek, "Run-Length-Limited Variable Length Coding with Error Propagation Limitation," U.S. Patent 3,689,899, 1972.
41. J. S. Eggenberger and P. Hodges, "Sequential Encoding and Decoding of Variable Word Length, Fixed Rate Data Codes," U.S. Patent 4,115,768, 1978.

Received November 5, 1980; revised March 5, 1981

The authors are located at the IBM General Products Division laboratory, 5600 Cottle Road, San Jose, California 95193.

AXD024638

EXHIBIT 17

Z80™-DMA
Z80A™-DMA

Zilog

Product Specification

OCTOBER 1977

6/12/77

PRELIMINARY

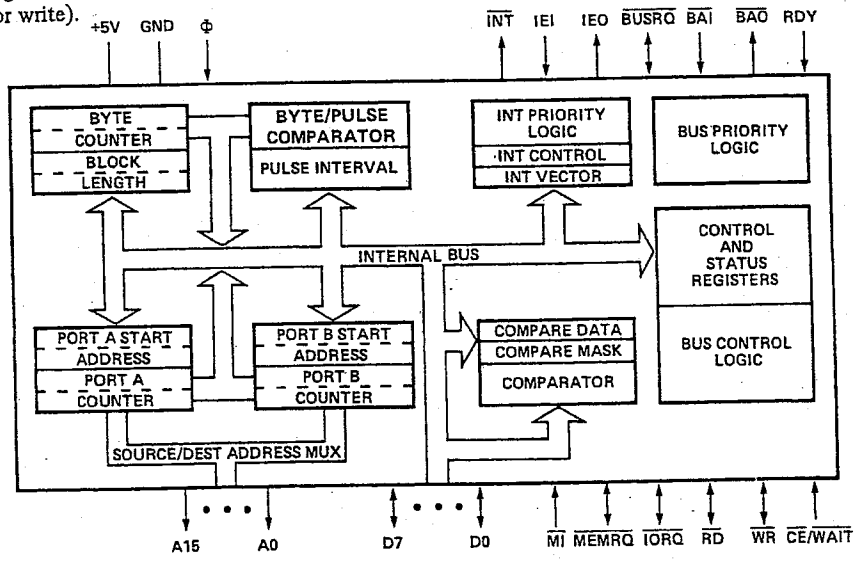
Zilog's Z80 microcomputer product line includes a third generation LSI component set, development systems and support software. The component set includes all the logic circuits necessary for the user to build high performance microcomputer systems with virtually no external logic and a minimal number of standard low-cost memory components. The Z80-DMA (Direct Memory Access) circuit is a programmable single-channel device which provides all address, timing and control signals to effect the transfer of blocks of data between two ports within a Z80-CPU based system. These ports may be either system main memory or any system peripheral I/O device. The DMA can also search a block of data for a particular byte (bit maskable), with or without a simultaneous transfer.

Structure

- N-channel Silicon Gate Depletion Load Technology
- 40 Pin DIP
- Single 5 volt supply
- Single phase 5 volt clock
- Single channel, two port

Features

- Three classes of operation:
  - –Transfer Only
  - –Search Only
  - –Search-Transfer
- Address and Block Length Registers fully buffered. Values for next operation may be loaded without disturbing current values.
- Dual addresses generated during a transfer (one for read port and one for write).
- Programmable data transfers and searches, automatically incrementing or decrementing the port addresses from programmed starting addresses (they can also remain fixed).
- Four modes of operation:
  - –Byte-at-a-time: One byte transferred per request
  - –Burst: Continues as long as ports are ready
  - –Continuous: Locks out CPU until operation complete
  - –Transparent: Steals refresh cycles
- Timing may be programmed to match the speed of any port.
- Interrupts on Match Found, End of Block, or Ready, may be programmed.
- An entire previous operation may be repeated automatically or on command. (Auto restart or Load)
- The DMA can signal when a specified number of bytes has been transferred, without halting transfer.
- Multiple DMA's easily configured for rotating priority.
- The channel may be enabled, disabled or reset under software control.
- Complete channel status upon program (CPU) request.
- Up to 1.25 megabyte Search or Transfer Rate.
- Daisy-chain priority interrupt and bus acknowledge included to provide automatic interrupt vectoring and bus request control, without need for additional external logic.
- TTL compatible inputs and outputs
- The CPU can read current Port counters, Byte counter, or Status Register. A mask byte can be set which defines which registers can be accessed during read operations.



DMA Internal Block Diagram

AX205541

Fig.1

TM Z80 is a trademark of Zilog, Inc

DMA Architecture

A block diagram of the Z80 DMA is shown in Figure 1. The internal structure consists of the following circuitry:

- *Bus Interface*: provides driver and receiver circuitry to interface to the Z80-CPU Bus.
- *Control Logic and Registers*: set the class, mode and other basic control parameters of the DMA.
- *Address, Byte Count and Pulse Circuitry*: generates the proper port addresses for the read and write operations, with provisions for incrementing or decrementing the address. When zero bytes remain to be handled, the byte count circuitry sets a flag in the status register. Pulse circuitry generates a pulse each time the byte counter lower 8-bits equal the pulse reg.
- *Timing Circuitry*: allows the user to completely specify the read/write timing for both of the channels' addressed ports.
- *Match Circuitry*: holds the match byte and a mask byte which allows for the comparison of only certain bits within the byte. If a match is encountered during a Search or Transfer, this circuitry sets a flag in the status register.
- *$\overline{INT}$ and $\overline{BUSRQ}$ Circuitry:* includes a control register which specifies the conditions under which the DMA can generate an interrupt; priority encoding logic to select between the generation of an $\overline{INT}$ or $\overline{BUSRQ}$ output under these conditions; and an interrupt vector register for automatic vectoring to the interrupt service routine.
- *Status Register*: holds current status of DMA.

Register Description

The following DMA-internal registers are available to the programmer:

- *Control Registers:* Hold DMA control information: such as, when to initiate an interrupt or pulse, what mode or class of operation to perform, etc. (Write Only) (8 Bits)
- *Timing Registers:* Hold read/write timing parameters for the two ports. (Write Only) (8 bits)
- *Interrupt Vector Register:* Holds the 8-bit vector that the DMA will put onto the data bus after receiving an $\overline{IORQ}$ during an interrupt acknowledge sequence if it is the highest priority device requesting an interrupt. (This register is readable only during interrupt acknowledge cycles.) (Read/Write) (8 bits)
- *Block Length Register:* Contains total block length of data to be searched and/or transferred. (Write Only) (16 bits)
- *Byte Counter:* Counts number of bytes transferred (or searched). On a Load or Continue the Byte Counter is reset to zero. Thereafter, each byte transfer operation increments it until it matches the contents of the Block Length Register, at which time End of Block is set in the status register and operation is suspended if programmed. Also if so programmed the DMA will generate an interrupt. (Read Only) (16 bits)
- *Compare Register:* Holds the byte for which a match is being sought in Search operations. (Write Only) (8 bits)
- *Mask Register:* Holds the 8 bit mask to determine which bits in the compare register are to be examined for a match. (Write Only) (8 bits)
- *Starting Address Registers (Port A and Port B):* Hold the starting addresses (upper and lower 8 bits) for the two ports involved in Transfer operations. In Search Only operations, only one port address would have to be specified. Only memory starting addresses require both upper and lower 8-bits; I/O ports are generally addressed with only the lower 8-bits, and in this case the address contained in the register is a generally fixed address. (Write Only) (16 bits each)
- *Address Counters (Port A and Port B):* These counters are loaded with the contents of the corresponding Starting Address Registers whenever Searches or Transfers are initiated with a Load or Continue. They are incremented, decremented or remain fixed, as programmed. (Read Only) (16 bits each)
- *Pulse Control Register:* Holds program-supplied length (in bytes) of block after which the DMA will provide a signal pulse on the $\overline{INT}$ pin. (Since this occurs while both $\overline{BUSRQ}$ and $\overline{BUSAK}$ are active, the CPU will not interpret this as an interrupt request. Instead, the signal is used to communicate with a peripheral I/O device.) (Write Only) (8 bits)
- *Status Register:* Match, End of Block, Ready Active, Interrupt Pending, and Write Address Valid bits indicate these functions when set. (Read Only) (8 bits)

Modes of Operation

The DMA may be programmed for one of four modes of operation. (See Command Byte 2B).

- *Byte at a time*: control is returned to the CPU after each one-byte cycle.
- *Burst*: operation continues as long as the DMA's RDY input is active, indicating that the relevant port is ready. Control returns to the CPU when RDY is inactive or at end of block or a match if so programmed.
- *Continuous*: the entire Search and/or Transfer of a block of data is completed before control is returned to CPU.
- *Transparent*: DMA operation occurs during normal memory refresh times without visible loss of CPU time.

Classes of Operation

The DMA has three classes of operation: Transfer only, Search Only and a combined Search-Transfer. (See Command Byte 1A.)

During a Transfer, data is first read from one port and then written to the other port, byte by byte. (The DMA's two ports are termed Port A and Port B.) The ports may be programmed to be either system main memory or peripheral I/O devices. Thus, a block of data might be written from a peripheral to another; or it might be written from one area in main memory to another; or from a peripheral to main memory.

During a Search, data is read only, and compared byte by byte against two DMA-internal registers, one of which contains a match byte and the other an optional mask byte which allows only certain bits to be compared. If any byte of searched data matches, a DMA-internal status bit is set; if programmed to do so, the DMA will then suspend operation and/or generate an interrupt.

The third class of operation is a combined Search-Transfer. In such an operation a block of data is transferred as described above until a match is found; then, as in a Search Only operation, the transfer may be suspended and/or an interrupt generated.

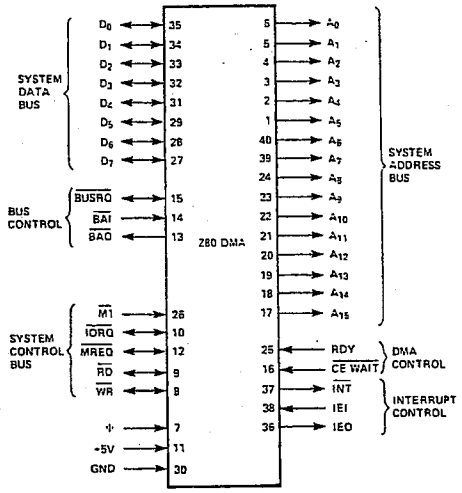
Addressing

The DMA's addressing of ports is either fixed or sequential, incrementing or decrementing from a starting address. The length of the operation (number of bytes) is specified by the programmed contents of a block length register. The DMA can address block lengths of up to 64K bytes. During a transfer two separate port addresses are generated, one during the Read cycle and one during the Write cycle.

Operating Sequence

Once the DMA has been programmed it may be "Enabled" (command byte 2d or 2a). In the enabled condition when Ready goes active the DMA will request the bus by bringing $\overline{BUSRQ}$ low. The CPU will acknowledge this with a $\overline{BUSACK}$ which will normally be attached to $\overline{BAI}$. When the DMA receives $\overline{BAI}$ it will start its programmed operation releasing $\overline{BUSRQ}$ to a "high" state when it is through.

Z80-DMA Pin Description



A_0–A_{15}	System Address Bus. All sixteen of these pins are used by the DMA to address system main memory or an I/O port (output)
D_0–D_7	System Data Bus. Commands from the CPU, DMA status and data from memory or peripherals are transferred on these tristate pins (input/output)
+5V	Power
GND	Ground
Φ	System clock (input)
$\overline{M1}$	Machine cycle One signal from CPU (input)
$\overline{IORQ}$	Input/Output Request to and from the System Bus (input/output)
$\overline{MREQ}$	Memory REQuest to the System Bus (input/output)
$\overline{RD}$	ReaD to and from the System Bus (input/output)
$\overline{WR}$	WRite to and from the System Bus (input/output)
$\overline{CE/WAIT}$	Chip Enable: may also be programmed to be $\overline{WAIT}$ during time when $\overline{BAI}$ is low (input)
$\overline{BUSRQ}$	BUS ReQuest. Requests control of the CPU Address Bus, Data Bus and Status/Control Bus (input/output, open drain)
$\overline{BAI}$	Bus Acknowledge In. Signals that the system buses have been released for DMA control (input)
$\overline{BAO}$	Bus Acknowledge Out. $\overline{BAI}$ and $\overline{BAO}$ form a daisy-chain connection for system-wide priority bus control (output)
$\overline{INT}$	INTerrupt request (output, open drain)
IEI	Interrupt Enable In (input)
IEO	Interrupt Enable Out. IEI and IEO form a daisy-chain connection for system-wide priority interrupt control (output)
RDY	ReaDY is monitored by the DMA to determine when a peripheral device associated with a DMA port is ready for a read or write operation (input, programmable as active high or low)

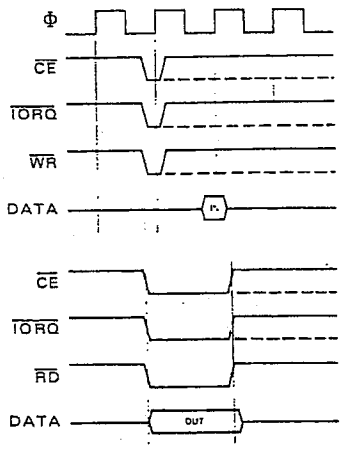
DMA Timing Waveforms

DMA Command Write Cycle

Illustrated here is the timing associated with a command byte or control byte being written to the DMA which is to be loaded into internal registers. Z80 Output instructions satisfy this timing.

DMA Register Read Cycle

This timing is used when a read operation is performed on the DMA to access the contents of the Status Register, Address Counter or other readable registers. Z80 Input instructions satisfy this timing.



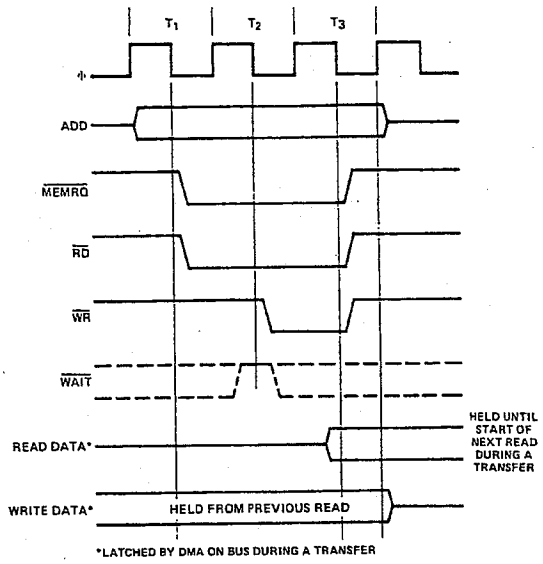
AX205543

DMA Timing Waveforms (Continued)

STD Memory Timing

This timing is exactly the same as used by the Z80-CPU to access system main memory, either in a Read or Write operation. The DMA will default to this timing after a power-on reset, or when a Reset or Reset Timing command is written to it; and unless otherwise programmed, will use this timing during all Transfer or Search operations involving system main memory. During the memory Read portion of a transfer cycle, data is latched in the DMA on the negative edge of Φ during T_3 and held into the following Write cycle. During the memory Write portion of a transfer cycle, data is held from the previous Read cycle and released at the end of the present cycle.

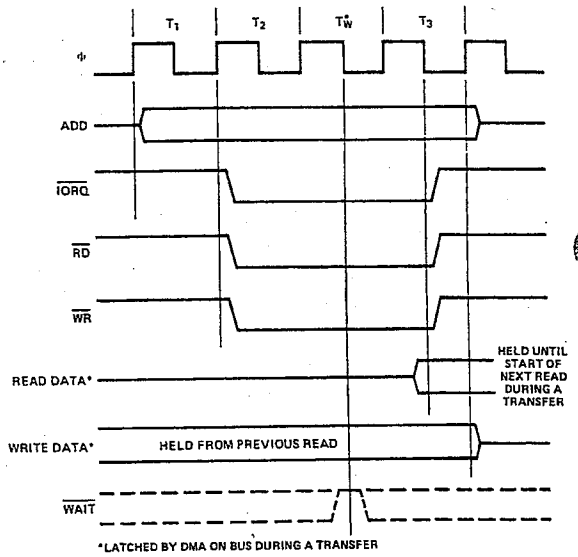
NOTE: The DMA is normally programmed for a 3 T-cycle duration in memory transactions. But $\overline{WAIT}$ is sampled during the negative transition of T_2, and if it is low, T_2 will be extended another T-cycle, during which $\overline{WAIT}$ will again be sampled. The duration of a memory transaction cycle may thus be indefinitely extended.



STD Peripheral Timing

This timing is identical to the Z80-CPU's Read/Write timing to I/O peripheral devices. The DMA will default to this timing after a power-on reset, or when a Reset or Reset Timing command is written to it; and unless otherwise programmed, will use this timing during all Transfer or Search operations involving I/O peripherals. During the I/O Read of a transfer cycle, data is latched on the negative edge of Φ during T_3 and is then held into the Write cycle. During an I/O Write, data is held from the previous Read cycle until the end of the Write cycle.

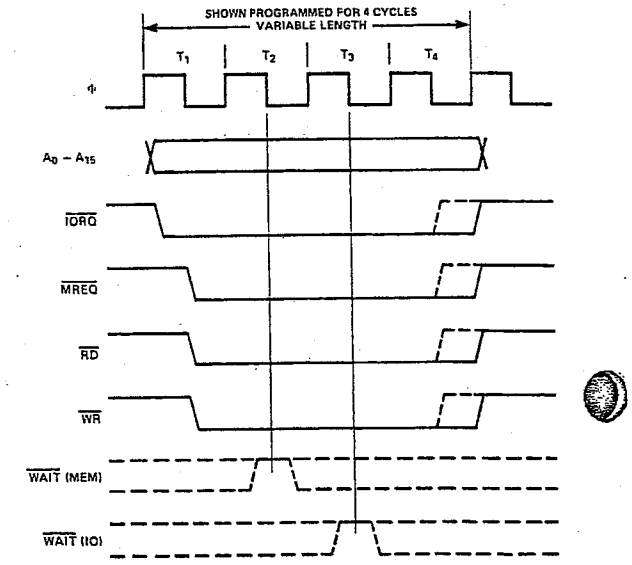
NOTE: If $\overline{WAIT}$ is low during the negative transition of T_W*, then T_W* will be extended another T-cycle and $\overline{WAIT}$ will again be sampled. The duration of a peripheral transaction cycle may thus be indefinitely extended.



Variable Cycle

The Variable feature of the DMA allows the user to program the DMA's memory or peripheral transaction timing to values different than given above in the standard default diagrams. This permits the designer to tailor his timing to the particular requirements of his system components, and maximizes the data transfer rate while eliminating external signal conditioning logic. Cycle length can be one to four T-cycles (more if $\overline{WAIT}$ is used). Signal timing can be varied as shown. During a transfer, data will be latched by the DMA on the clock edge causing the rising edge of $\overline{RD}$ and will be held on the data lines until the end of the following Write cycle.

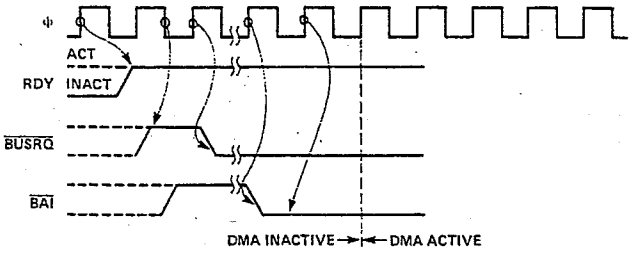
(See Timing Control Byte, page 7.)



DMA Timing Waveforms (Continued)

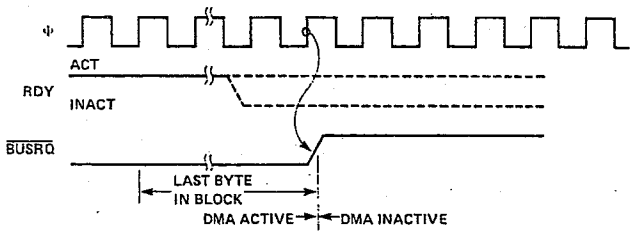
DMA Bus Request and Acceptance for Byte-at-a-Time, Burst, and Continuous Mode

Ready is sampled on every rising edge of Φ. When it is found to be active, the following rising edge of Φ generates $\overline{BUSRQ}$. After receiving $\overline{BUSRQ}$ the CPU will grant a $\overline{BUSAK}$ which will be connected to $\overline{BAI}$ either directly or through the Bus Acknowledge Daisy Chain. When a low is detected on $\overline{BAI}$ (sampled on every rising edge of Φ), the next rising edge of Φ will start an active DMA cycle.



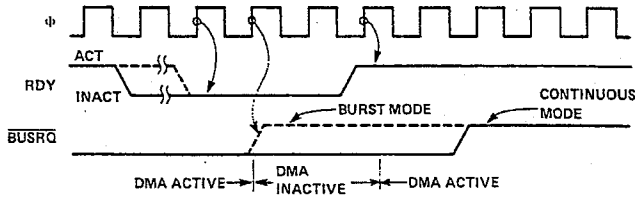
DMA Bus Release at End of Block for Burst or Continuous Mode

Timing for End of Block and DMA not programmed for Auto-restart.



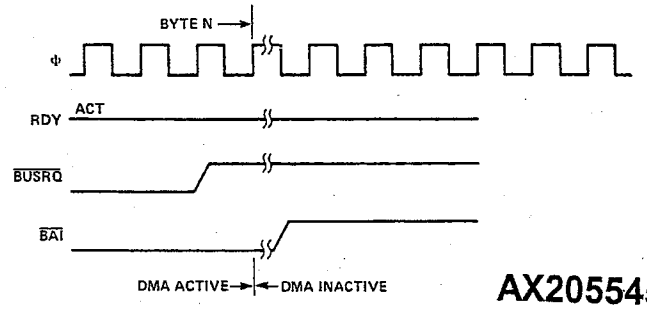
DMA Bus Release with 'Ready' for Burst and Continuous Mode

The DMA will relinquish the bus after RDY has gone inactive (Burst mode) or after an End of Block or a Match is found (Continuous mode). With RDY inactive, the DMA in Continuous mode is inactive but maintains control of the bus ($\overline{BUSRQ}$ low) until the cycle is resumed when RDY goes active.



DMA Bus Release for Byte-at-a-Time Mode

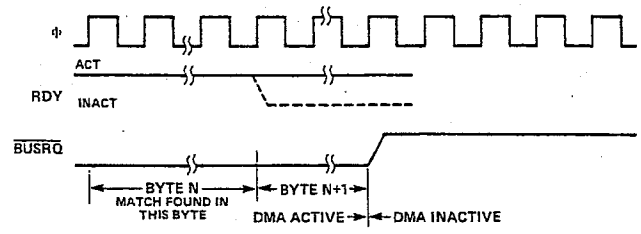
In the Byte mode the DMA will release $\overline{BUSRQ}$ on the rising edge of Φ prior to the end of each Read cycle in Search Only or each Write cycle in a Transfer, regardless of the state of RDY. The next bus request will come after both $\overline{BUSRQ}$ and $\overline{BAI}$ have returned high.



AX205545

DMA Bus Release with Match for Burst or Continuous Modes

When a Match is found and the DMA is programmed to stop on Compare, the DMA performs an operation on the next byte and then releases bus.



Reading from the DMA Internal Registers

Seven registers are available on the DMA for reading. They are: 8 bits of the status register, the upper and lower 8 bits of the block length register, and two port address registers.

These are available to be read sequentially: status, BLK Lower, BLK Upper, Port A Address lower, Port A Address Upper, Port B Address lower, Port B Address upper. An internal pointer points to each register in turn as each READ is accomplished. If a register is not to be read, it may be excluded by programming a 0 in the Read Byte. The internal pointer will skip any register not programmed with a 1 in the Read Byte. After a Reset or a Load, Reset RD must be given to set the internal pointer pointing to the first register programmed to be read by the Read Byte. After RD Status, the pointer will be pointing to the status register regardless of the Read Byte and the next read will be from the status register. The following read will be from the register pointed to before RD Status.

Programming the DMA

Previous sections of this specification have indicated the various functions and modes of the DMA. The diagrams and charts below will show how the DMA is programmed to select among these functions and modes and to adapt itself to the requirements of the user system. More detailed programming information is available in the *Z80-DMA Technical manual.*

The Z80-DMA chip may be in an "enable" state, in which it can gain control of the system buses and direct the transfer of data between its ports, or in a "disable" state, when it cannot gain control of the bus. Program commands can be written to it in either state, but writing a command to it automatically puts it in the disable state, which is maintained until an enable command is issued to the DMA. The CPU must program it in advance of any data search or transfer by addressing it as an I/O port and sending it a sequence of 8 bit command bytes via the system data bus using Output instructions. When the DMA is powered up or reset by any means, the DMA will automatically be placed into a disable state, in which it can initiate neither bus requests nor data transfers nor interrupts.

The command bytes contain information to be loaded into the DMA's control and other registers and/or information to alter the state of the chip, such as an Enable Interrupt command. The command structure is designed so that certain bits in some commands can be set to alert the DMA to expect the next byte written to it to be for a particular internal register.

The following diagrams and charts give the function of each bit in the six different command bytes. Two of these are defined as being from Group 1, and are termed command bytes 1A and 1B. These Group 1 commands contain the most basic DMA set-up information. The other four are categorized as Group 2, and are termed commands 2A, 2B, 2C and 2D. Group 2 words specify more detailed set-up information.

Command Byte 1A

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	BLOCK LENGTH (UPPER) FOLLOWS	BLOCK LENGTH (LOWER) FOLLOWS	PORT A STARTING ADDRESS (UPPER) FOLLOWS	PORT A STARTING ADDRESS (LOWER) FOLLOWS	SOURCE PORT	CLASS CONTROL C_1	CLASS CONTROL C_0

Specifies Group 1 — Byte 1A cannot be 00

C_1	C_0	Function
0	0	Not allowed. (Command Byte 1B)
0	1	Transfer Only.
1	0	Search Only.
1	1	Search and Transfer.

$D_2 = 1$ Port A is read from, Port B is written to (unless the Search Only Mode has been selected, in which case Port B is never addressed).

$D_2 = 0$ Port B is read from, Port A is written to (unless the Search Only Mode has been selected, in which case Port A is never addressed).

Command Byte 1B

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	TIMING BYTE FOLLOWS	ADDRESS FIXED	ADDRESS INCREMENTS DECREMENTS	I/O OR MEMORY	PORT A OR B	0	0

Specifies Group 1 — Specifies Byte 1B

$D_4 = 1$ Address for this port increments after each byte.

$D_4 = 0$ Address for this port decrements after each byte.

$D_3 = 1$ This port addresses an I/O peripheral.

$D_3 = 0$ This port addresses main memory.

$D_2 = 1$ This word programs Port A.

$D_2 = 0$ This word programs Port B.

AX205546

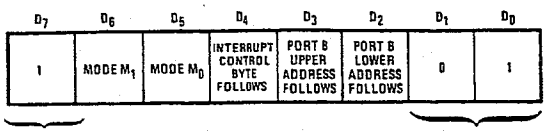
Command Byte 2A

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	ENABLE CHIP	ENABLE INTERRUPT	MATCH BYTE FOLLOWS	MASK BYTE FOLLOWS	STOP ON COMPARE	0	0

Specifies Group 2 — Specifies Byte 2A

Programming the DMA (Continued)

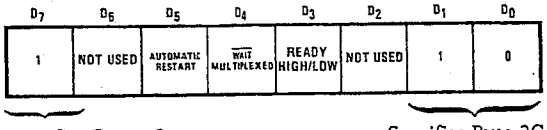
Command Byte 2B



Specifies Group 2 — Specifies Byte 2B

M_1	M_0	Mode
0	0	Byte
0	1	Continuous
1	0	Burst
1	1	Transparent

Command Byte 2C



Specifies Group 2 — Specifies Byte 2C

$D_5 = 1$ Automatically repeats entire operation when end of block is reached.

$D_5 = 0$ No affect.

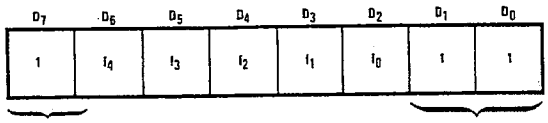
$D_4 = 1$ $\overline{CE}$ and $\overline{WAIT}$ multiplexed on same pin.

$D_4 = 0$ $\overline{CE}$ only.

$D_3 = 1$ Ready active high.

$D_3 = 0$ Ready active low.

Command Byte 2D



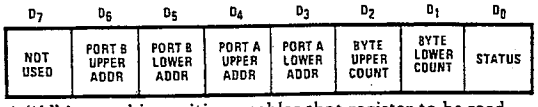
Specifies Group 2 — Specifies Byte 2D

Hex	f_4	f_3	f_2	f_1	f_0	
C3	1	0	0	0	0	Reset
C7	1	0	0	0	1	Reset Port A Timing
CB	1	0	0	1	0	Reset Port B Timing
CF	1	0	0	1	1	Load
D3	1	0	1	0	0	Continue
AB	0	1	0	1	0	Enable Int
AF	0	1	0	1	1	Disable Int
A3	0	1	0	0	0	Reset Int
87	0	0	0	0	1	Enable DMA
83	0	0	0	0	0	Disable DMA
BB	0	1	1	1	0	Read Byte Follows
A7	0	1	0	0	1	Reset RD
BF	0	1	1	1	1	RD Status
B3	0	1	1	0	0	Force Ready
B7	0	1	1	0	1	Enable After RETI
8B	0	0	0	1	0	Reset Status

Command Byte 2D Summary

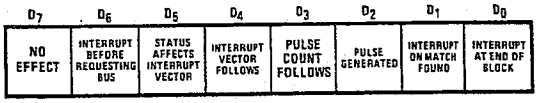
Reset	Resets all interrupt circuitry, disables interrupts and bus req. logic.
Reset Timing A or B:	Resets timing for Port A or B to standard Z80-CPU timing.
Load:	Zeros Byte Counter and loads Starting Address for both Ports.
Continue:	Resets byte counter only. Addresses continue from present location.
Enable Interrupt:	Permits interrupt to occur.
Disable Interrupt:	Inhibits interrupt from occurring.
Reset Interrupt:	Resets and disables all interrupt circuits (similar to RETI).
Enable DMA, Disable DMA:	Overall enable or disable for all operations except interrupts: does not reset any functions.
Read Byte Follows:	Next write to DMA will contain a mask to program which readable registers are to be read.
Reset RD:	Next read will be from 1st register set as readable by response mask.
RD Status:	Next read will be from status register.
Force Ready:	Ready will be considered active regardless of the state of external RDY pin. Used for Mem-Mem operations where no RDY signal is needed.
Enable after RETI:	DMA will not request bus until after it has received a RETI.
RST Status:	Resets Match and End of Block status bits.

Read Byte



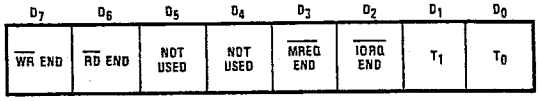
A "1" in any bit position enables that register to be read.

Interrupt Control Byte



A "1" in a bit position selects the option.

Timing Control Byte



T_1	T_0	Cycle Length
0	0	4
0	1	3
1	0	2
1	1	1

AX205547

A "0" in D_2, D_3, D_6, or D_7 will cause the corresponding control signal to end ½ clock time before the end of the cycle. Note: the total operation (Read and Write in Transfer or Read in Search) must be at least 2 cycles long.

Programming the DMA (Continued)

Mask Byte

A zero in a given bit position will cause a compare to be performed between that bit position in the compare word register and the same bit position in the data being read.

Match Byte

Up to an 8-bit word to be compared to $D_0 - D_7$ during a read. See MASK BYTE.

Status Byte (Status Bits Active–Low)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
NOT USED	NOT USED	END OF BLK	MATCH	INT. PENDING	NOT USED	READY ACTIVE	WRITE ADDRESS START

Pulse Count

This 8-bit word is loaded into a register. At the completion of each operation, the register is compared with the lower 8-bits of the byte counter. When it compares, the $\overline{INT}$ line is pulsed (but no interrupt is generated).

Interrupt Vector

This 8-bit byte is supplied to the CPU during Interrupt acknowledge if the DMA is the highest priority interrupting device.

If bit 5 of the Interrupt Control Byte (see p. 7) has been set and the DMA has been programmed to interrupt on a given status condition then D_1 and D_2 of the vector will be modified as follows:

Vector Bits D_2	D_1	
0	0	INT on RDY
0	1	Match
1	0	End of Blk
1	1	Match, End of Blk

DMA Programming Example

The following example will show how the DMA may be programmed to transfer data from a peripheral (Port A) to memory (Port B). The table of bytes may be stored in memory and transferred to the DMA with an output instruction such as an OTIR.

Port A	Data Flow	Port B
Peripheral Address "X"5H	→ Block Length 1000H Bytes	Memory Starting Address 1050H

READY from the peripheral is active high
Memory address increments on each write

		D_7	D_6	D_5	D_4	D_3	D_2	D_1 D_0	HEX
1	Command Byte 1a Sets the DMA to receive Block length and Port A address and sets direction of transfer	0 Group 1	1 Blk Length Upper Follows	1 Blk Length Lower Follows	0 No Port A Upper Addr Follows	1 Port A Lower Addr Follows	1 A→B	0 1 — 1a Transfer No Search	6D
2	Port A Address Lower 8-bits	0	0	0	0	0	1	0 1	01
3	Block Length Lower 8-bits	0	0	0	0	0	0	0 0	00
4	Block Length Upper 8-bits	0	0	0	1	0	0	0 0	10
5	Command Byte 1b Defines Port A as peripheral with fixed addresses	0 Group 1	0 No Timing Follows	1 Fixed Addresses	X	1 Port is IO	1 This is Port "A"	0 0 — 1b	
6	Command Byte 1b Defines Port B as a memory with incrementing addresses	0 Group 1	0 No Timing Follows	0 Address Changes	1 Address Increments	0 Port is Memory	0 This is Port "B"	0 0 — 1b	14
7	Command Byte 2b Sets mode to burst, sets DMA to expect Port B starting address	1 Group 2	1 0 — Burst Mode		0 No Int Cont Byte Follows	1 Port B Upper Addr Follows	1 Port B Lower Addr Follows	0 1 — 2b	CD
8	Port B Address Lower 8-bits	0	1	0	1	0	0	0 0	50
9	Port B Address Upper 8-bits	0	0	0	1	0	0	0 0	10
10	Command Byte 2c Sets Ready Active High	1 Group 2	X	0 No Auto Restart	0 No wait States	1 Rdy Active High	X	1 0 — 2c	
11	Command Byte 2d loads starting addresses and resets block counter	1 Group 2	1	0	0	1	1 — Load	1 1 — 2A	CF
12	Command Byte 2d Enables DMA to start operation	1 Group 2	0	0	0	0	1 — ENABLE DMA	1 1 — 2d	87

To reload the same addresses and block length for a subsequent operation, only two bytes are needed.

1. Command byte 2d — 11001111 — CF
   Reloads port addresses and block length — Load
2. Command byte 2d — 10001011 — 87
   Enables DMA — Enable DMA

AX205548

Absolute Maximum Ratings

Temperature Under Bias	Specified operating range.
Storage Temperature	−65°C to +150°C
Voltage On Any Pin with Respect to Ground	−0.3V to +7V
Power Dissipation	1.5W

Note: All AC and DC characteristics remain the same for the military grade parts except I_{CC}.

I_{CC} = 200 mA.

*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Z80-DMA D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	−0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}−.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	−0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −250 µA
V_{CC}	Power Supply Current			150	mA	t_C = 400 nsec
I_{LI}	Input Leakage Current			10	µA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	µA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			−10	µA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	µA	0 < V_{IN} < V_{CC}

Z80A-DMA D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	−0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}−.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	−0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −250 µA
V_{CC}	Power Supply Current		90	200	mA	t_C = 250nsec
I_{LI}	Input Leakage Current			10	µA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	µA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			−10	µA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	µA	0 < V_{IN} < V_{CC}

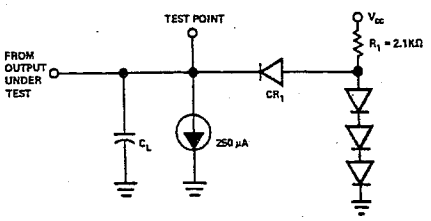
AX205549

Capacitance

T_A = 25°C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C_Φ	Clock Capacitance	35	pF	Unmeasured Pins Returned to Ground
C_{IN}	Input Capacitance	5	pF	
C_{OUT}	Output Capacitance	10	pF	

Load Circuit for Output

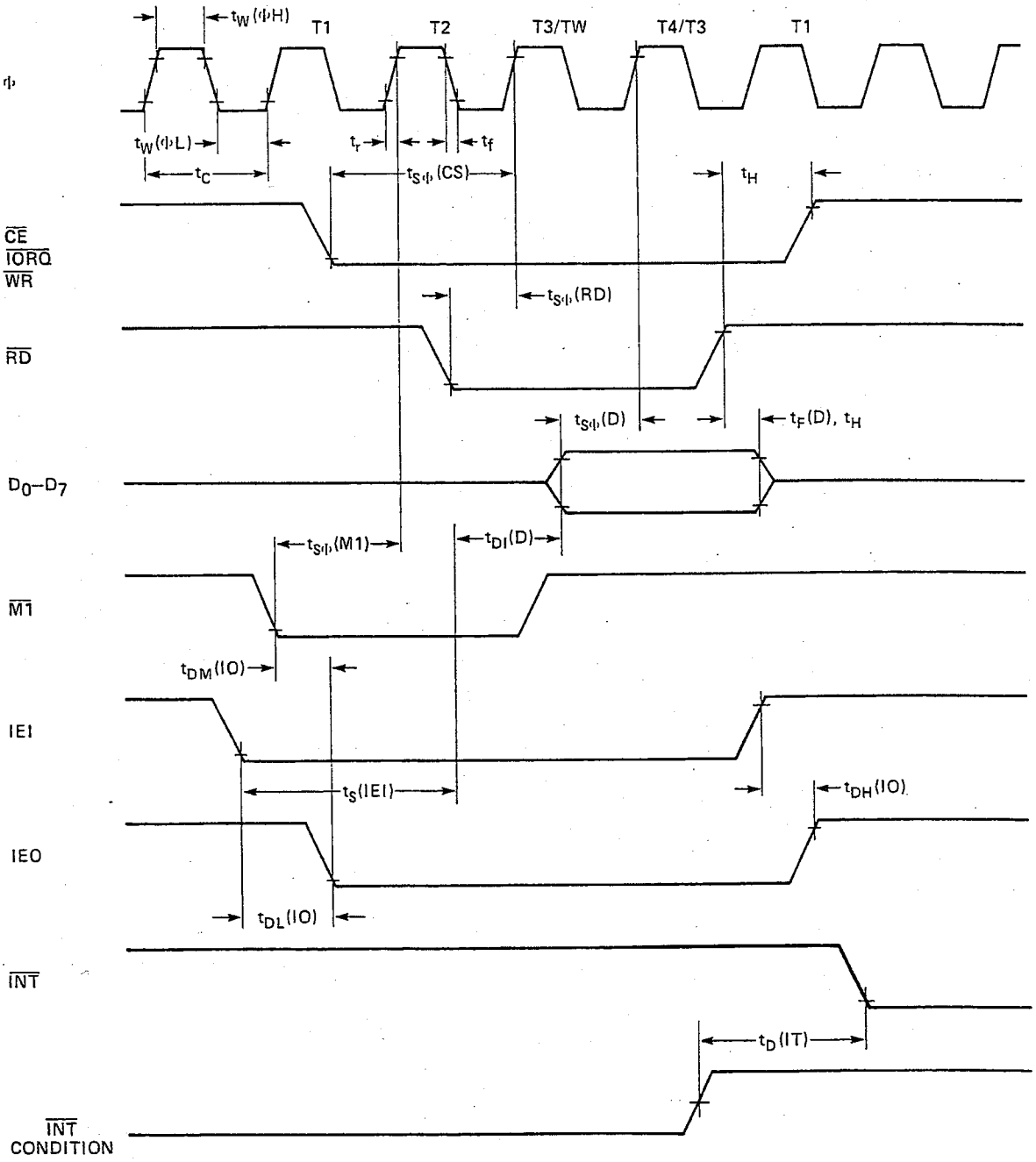


A.C. Timing Diagrams

Z80 and Z80A as a Peripheral Device (Inactive State)

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	ΔV	= +0.5V



AX205550

A.C. Characteristics Z80-DMA

Z80-DMA as a Peripheral Device (Inactive State).
T_A = 0°C to 70°C, Vcc = +5V±5%, Unless Otherwise Noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c $t_{w(\Phi H)}$ $t_{w(\Phi L)}$ t_r, t_f	Clock Period Clock Pulse Width, Clock High Clock Pulse Width, Clock Low Clock Rise and Fall Times	400 170 170	[1] 2000 2000 30	nsec nsec nsec nsec	
	t_H	Any Hold Time for Specified Setup Time	0		nsec	
$\overline{CE}$, $\overline{WR}$ $\overline{IORQ}$	$t_{S\Phi(CS)}$	Control Signal Setup Time to Rising Edge of Φ During Write Cycle ($\overline{IORQ}$, $\overline{WR}$, $\overline{CE}$)	280		nsec	
D_{0-7}	$t_{DR(D)}$ $t_{S\Phi(D)}$ $t_{DI(D)}$ $t_{F(D)}$	Data Output Delay from Falling Edge of $\overline{RD}$ Data Setup Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle Delay to Floating Bus (Output Buffer Disable Time)	50	430 340 160	nsec nsec nsec nsec	[2] C_L = 50pF [3]
IEI	$t_{S(IEI)}$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	$t_{DH(IO)}$ $t_{DL(IO)}$ $t_{DM(IO)}$	IEO Delay Time from Rising Edge of IEI IEO Delay Time from Falling Edge of IEI IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		210 190 300	nsec nsec nsec	C_L = 50pF
$\overline{M1}$	$t_{S\Phi(M1)}$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle. See Note B.	210		nsec	
$\overline{RD}$	$t_{S\Phi(RD)}$	$\overline{RD}$ Setup Time to Rising Edge of Φ During $\overline{M1}$ Cycle	240		nsec	
$\overline{INT}$	$t_{D(IT)}$	$\overline{INT}$ Delay Time from Condition Causing $\overline{INT}$. $\overline{INT}$ generated only when DMA is inactive.		500	nsec	
BAO	$t_{DH(BO)}$ $t_{DL(BO)}$	BAO Delay from Rising Edge of BAI BAO Delay from Falling Edge of BAI	150 150	200 200	nsec nsec	

[1] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

A. $2.5\,t_c > (N-2)\,t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

AX205551

A.C. Characteristics Z80A-DMA

Z80A-DMA as a Peripheral Device (Inactive State).
T_A = 0°C to 70°C, Vcc = +5V±5%, Unless Otherwise Noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	250	[1]	nsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	105	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	105	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Times		30	nsec	
	t_H	Any Hold Time for Specified Setup Time	0		nsec	
$\overline{CE}$	$t_{S\Phi(CS)}$	Control Signal Setup Time to Rising Edge of Φ During Write Cycle	145		nsec	
D_{0-7}	$t_{DR(D)}$	Data Output Delay from Falling Edge of $\overline{RD}$		380	nsec	[2]
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50	-	nsec	C_L = 50pF
	$t_{DI(D)}$	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle		250	nsec	[3]
	$t_{F(D)}$	Delay to Floating Bus (Output Buffer Disable Time)		110	nsec	
IEI	$t_{S(IEI)}$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	$t_{DH(IO)}$	IEO Delay Time from Rising Edge of IEI		160	nsec	
	$t_{DL(IO)}$	IEO Delay Time from Falling Edge of IEI		130	nsec	
	$t_{DM(IO)}$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		190	nsec	C_L = 50pF
$\overline{M1}$	$t_{S\Phi(M1)}$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle. See Note B.	90		nsec	
$\overline{RD}$	$t_{S\Phi(RD)}$	$\overline{RD}$ Setup Time to Rising Edge of Φ During $\overline{M1}$ Cycle	115		nsec	
$\overline{INT}$	$t_{D(IT)}$	$\overline{INT}$ Delay Time from Condition Causing $\overline{INT}$. $\overline{INT}$ generated only when DMA is inactive.		500	nsec	
BAO	$t_{DH(BO)}$	BAO Delay from Rising Edge of BAI	150	200	nsec	
	$t_{DL(BO)}$	BAO Delay from Falling Edge of BAI	150	200	nsec	

[1] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

A. $2.5\ t_c > (N-2)\ t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

AX205552

A.C. Characteristics Z80-DMA

Z80-DMA as a Bus Controller (Active State)
$T_A = 0°C$ to $70°C$, Vcc = +5V±5%, Unless Otherwise Noted.

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	4	[12]	μsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	180	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	180	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		145	nsec	
	$t_{F(AD)}$	Delay to Float		110	nsec	C_L = 50pF
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[1]		nsec	D
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec	D
	t_{ca}	Address Stable from $\overline{RD}$ or $\overline{WR}$	[3]		nsec	D
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec	D
D_{0-7}	$t_{D(D)}$	Data Output Delay		260	nsec	
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During Read When Rising Edge Ends $\overline{RD}$	50		nsec	C_L = 200pF
	$t_{S\overline{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During Read When Falling Edge Ends $\overline{RD}$	60		nsec	
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec	D
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec	D
	t_{cdf}	Data Stable From $\overline{WR}$	[7]		nsec	D
	t_H	Any Hold Time for Setup Time	0		nsec	
$\overline{MREQ}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay from Rising Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_{DL\Phi(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	C_L = 50pF
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec	D
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec	D
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ Low		90	nsec	
	$t_{DL\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay from Falling Edge of Clock, $\overline{IORQ}$ Low		110	nsec	
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ High		100	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay from Falling Edge of Clock, $\overline{IORQ}$ High		110	nsec	
$\overline{RD}$	$t_{DL\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ Low		100	nsec	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ Low		130	nsec	
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ High		100	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ High		110	nsec	
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock, $\overline{WR}$ Low		80	nsec	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ Low		90	nsec	
	$t_{DH\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ High		100	nsec	C_L = 50pF
	$t_{DH\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock, $\overline{WR}$ High		100	nsec	
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec	
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{BUSRQ}$	$t_{D(BQ)}$	$\overline{BUSRQ}$ Delay Time from Rising Edge of Clock	100		nsec	
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		100	nsec	

[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$
[2] $t_{aci} = t_c - 80$
[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$
[4] $t_{caf} = t_{w(\Phi L)} + t_r - 60$

[5] $t_{dcm} = t_c - 180$
[6] $t_{dci} = t_{w(\Phi L)} + t_r - 180$
[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 50$

[8] $t_{w(MRL)} = t_c - 40$
[9] $t_{w(MRH)} = t_c - 40$ Std. CPU Timing
$t_{w(MRH)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

[10] $t_{w(WR)} = t_c - 40$ Std. CPU Timing
$t_{w(WR)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

NOTES:
A. Data should be enabled onto the DMA data bus when $\overline{RD}$ is active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V±5%
(1) ΔC_L = +100pF (A_0–A_{15} and Control Signals), add 30 nsec to timing shown.
D. During Standard CPU Timing

AX205553

A.C. Characteristics Z80A-DMA

Z80A-DMA as a Bus Controller (Active State)
$T_A = 0°C$ to $70°C$, Vcc = +5V±5%, Unless Otherwise Noted.

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	.25	[12]	μsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	110	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	110	2000	nsec	
	t_r, t_f	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		110	nsec	
	$t_{F(AD)}$	Delay to Float		90	nsec	C_L = 50pF
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[1]		nsec	D
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec	D
	t_{ca}	Address Stable from $\overline{RD}$ or $\overline{WR}$	[3]		nsec	D
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec	D
D_{0-7}	$t_{D(D)}$	Data Output Delay		180	nsec	
	$t_{F(D)}$	Delay to Float During Write Cycle			nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During Read When Rising Edge Ends $\overline{RD}$	35		nsec	C_L = 200pF
	$t_{S\overline{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During Read When Falling Edge Ends $\overline{RD}$	50		nsec	
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec	D
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec	D
	t_{cdf}	Data Stable From $\overline{WR}$	[7]		nsec	D
	t_H	Any Hold Time for Setup Time		0	nsec	
$\overline{MREQ}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ Low		75	nsec	
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay from Rising Edge of Clock, $\overline{MREQ}$ High		75	nsec	
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ High		75	nsec	C_L = 50pF
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec	D
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec	D
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ Low		75	nsec	
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ High		80	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay from Falling Edge of Clock, $\overline{IORQ}$ High		80	nsec	
$\overline{RD}$	$t_{DL\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ Low		75	nsec	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ Low		95	nsec	
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ High		75	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ High		80	nsec	
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock, $\overline{WR}$ Low		60	nsec	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ Low		80	nsec	
	$t_{DH\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ High		80	nsec	C_L = 50pF
	$t_{DH\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock, $\overline{WR}$ High		80	nsec	
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec	
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{BUSRQ}$	$t_{D(BQ)}$	$\overline{BUSRQ}$ Delay Time from Rising Edge of Clock	100		nsec	
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		80	nsec	

[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$
[2] $t_{aci} = t_c - 80$
[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$
[4] $t_{caf} = t_{w(\Phi L)} + t_f - 60$

[5] $t_{dcm} = t_c - 180$
[6] $t_{dci} = t_{w(\Phi L)} + t_r - 180$
[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 50$

[8] $t_{w(MRL)} = t_c - 40$
[9] $t_{w(MRH)} = t_c - 40$ Std. CPU Timing
$t_{w(MRH)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

[10] $t_{w(WR)} = t_c - 40$ Std. CPU Timing
$t_{w(WR)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

NOTES:
A. Data should be enabled onto the DMA data bus when $\overline{RD}$ is active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V±5%
(1) ΔC_L = +100pF(A_0–$A_{15}$and Control Signals), add 30 nsec to timing shown.
D. During Standard CPU Timing

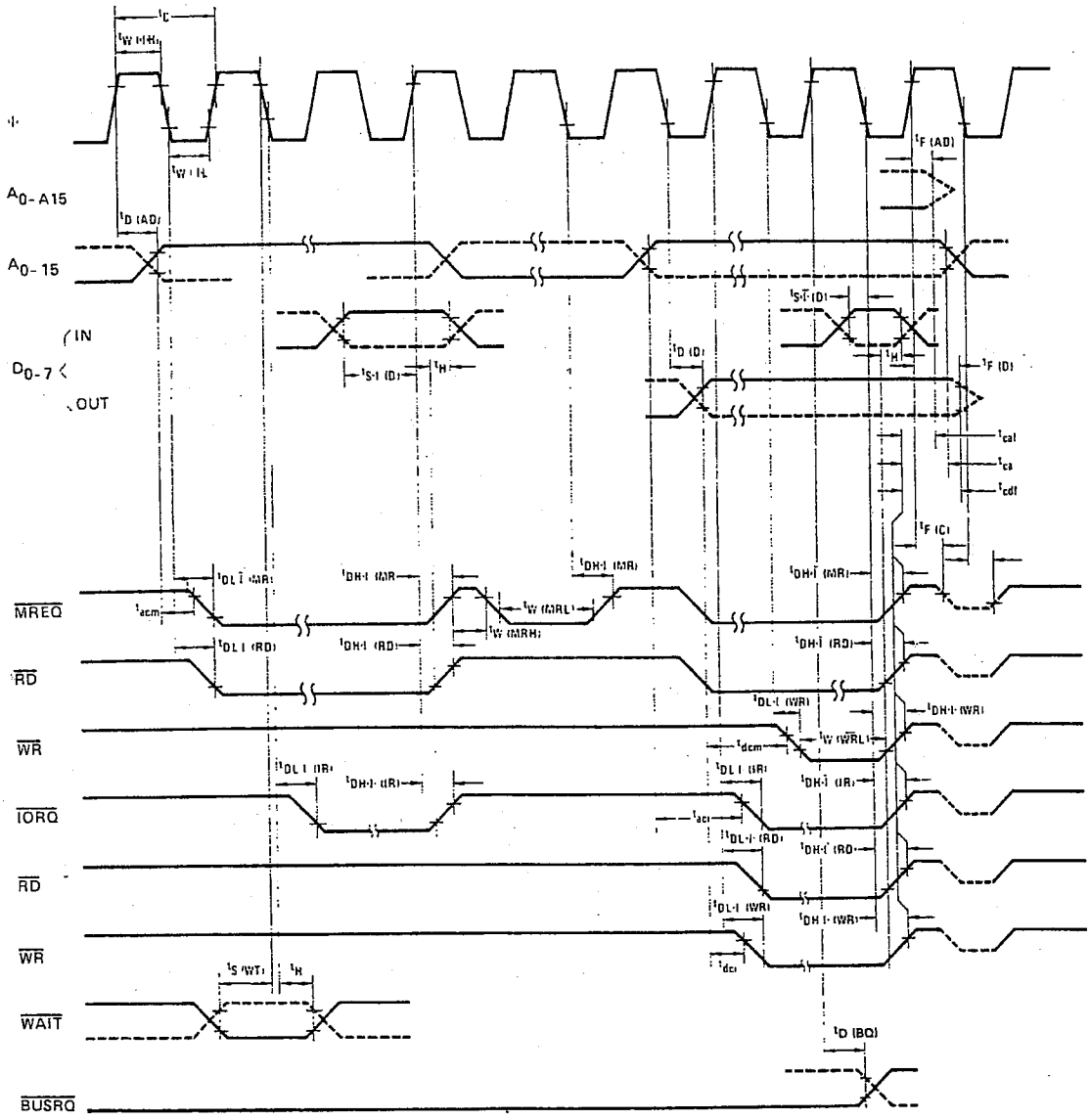
AX205554

A.C. Timing Diagrams

Z80 and Z80A as a Bus Controller (Active State)

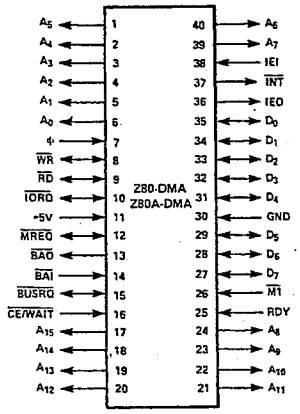
Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	ΔV =	±0.5V

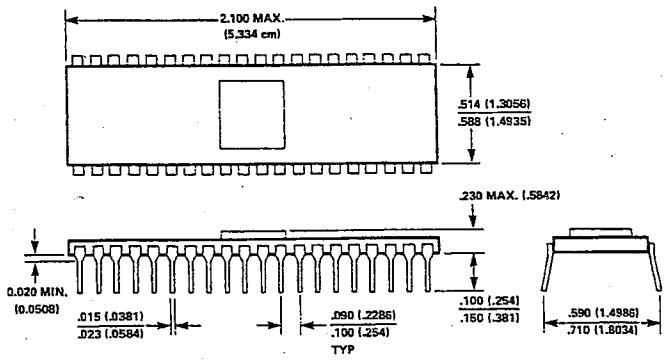


AX205555

Package Configuration



Package Outline



*Dimensions for metric system are in parentheses

ZILOG U.S. DISTRIBUTORS

EASTERN

Hallmark Electronics
4739 Commercial Drive
Huntsville, AL 35805
TEL 205 837 8700
TWX 810 726 2187

Hallmark Electronics
1302 West McNab Road
Fort Lauderdale, FL 33309
TEL 305 971 9280
TWX 510 956 9720

Hallmark Electronics
7233 Lake Ellenor Drive
Orlando, FL 32809
TEL 305 855 3020
TWX 810 850 0183

Hallmark Electronics
3355 Amberton Drive
Baltimore MD 21227
TEL 301 796 9300
TWX 710 862 1942

Hallmark Electronics
1208 Front Street
Building K
Raleigh, NC 27609
TEL 919 832 4465
TWX 510 928 1831

Hallmark Electronics
Pike Industrial Park
Huntington Valley, PA
TEL 215 355 7300
TWX 510 667 1750

Summit
916 Main Street
Buffalo, NY 14202
TEL 716 884 3450

Wilshire Electronics
2554 State Street
Hamden, CT 06517
TEL 203 281 1166
TWX 800 922 1734

Wilshire Electronics
1855 New Highway
Farmingdale, LI, NY 11735
TEL 516 293 5775
TWX 212 895 8707

Wilshire Electronics
One Wilshire Road
Burlington, MA 01803
TEL 617 272 8200
TWX 710 332 6359

Wilshire Electronics
1111 Paulison Avenue
Clifton, NJ 07015
TEL 201 340 1900
TWX 710 989 7052

MIDWESTERN

Hallmark Electronics
180 Grossen Avenue
Elk Grove Village, IL 60076
TEL 312 437 8800
TWX 910 223 3645

Hallmark Electronics
11870 West 91st Street
Congleton Industrial Park
Shawnee Mission, KS 66214
TEL 913 888 4747
TWX 910 749 6620

Hallmark Electronics
9201 Penn Avenue South
Suite 10
Bloomington, MN 55435
TEL 612 884 9056
TWX 910 576 3187

Hallmark Electronics
13789 Rider Trail
Earth City, MO 63045
TEL 314 291 5350
TWX 910 760 0671

Hallmark Electronics
6969 Worthington Galena Road
Worthington, OH 43085
TEL 614 846 1882

Hallmark Electronics
4846 S. 83rd E. Avenue
Tulsa, OK 74145
TEL 918 835 8458
TWX 910 845 2290

Hallmark Electronics
3100-A Industrial Terrace
Austin, TX 78758
TEL 512 837 2841
TWX 910 874 2031

Hallmark Electronics
9333 Forest Lane
Dallas, TX 75222
TEL 214 234-7400
TWX 910 867 4721

Hallmark Electronics
8000 W. Glenn
Houston, TX 77063
TEL 713 781 6100
TWX 910 881 2711

Hallmark Electronics
237 South Curtis
West Allis, WI 53214
TEL 414 476 1270
TWX 910 262 3186

RM Electronics
4860 South Division
Kentwood, MI 49508
TEL 616 531 9300
TWX 810 273 8779

MOUNTAIN

Century Electronics
121 Elizabeth, NE
Albuquerque, NM 87123
TEL 505 292 2700
TWX 910 989 0625

Century Electronics
2150 South 300 West
Salt Lake City, UT 84115
TEL 801 487 8551
TWX 910 925 5686

Century Electronics
8155 West 48th Avenue
Wheatridge, CO 80033
TEL 303 424 1985
TWX 910 938 0393

R. V. Weatherford Co.
3905 South Mariposa
Englewood, CO 80110
TEL 303 761 5432
TWX 910 933 0173

WESTERN

Intermark Electronics
1802 E. Carnegie Avenue
Santa Ana, CA 92705
TEL 714 540 1322
TWX 910 595 1583

Intermark Electronics
4040 Sorrento Valley Blvd.
San Diego, CA 92121
TEL 714 279 5200
714 453 9005
TWX 910 335 1515

Intermark Electronics
1020 Stewart Drive
Sunnyvale, CA 94086
TEL 408 738 1111
TWX 910 339 9312

R.V. Weatherford Co.
6921 San Fernando Road
Glendale, CA 91201
TEL 213 849 3451
TWX 910 498 2223

R.V. Weatherford Co.
1550 Babbitt Avenue
Anaheim, CA 92805
TEL 714 634 9600
TWX 910 593 1334

R.V. Weatherford Co.
1095 East Third Street
Pomona, CA 91765
TEL 714 623 1261
TWX 910 581 3811

R.V. Weatherford Co.
3240 Hillview Avenue
Stanford Industrial Park
Palo Alto, CA 94304
TEL 415 493 5373

R.V. Weatherford Co.
3311 W. Earll Drive
Phoenix, AZ 85017
TEL 602 272 7144
TWX 910 951 0636

Sterling Electronics
5608 6th Avenue South
Seattle, WA 98108
TEL 206 762 9100
TLX 32-9652

CANADA

Future Electronics
5647 Ferrier Street
Montreal, Quebec,
CANADA H4P 2K5

Ordering Information

C – Ceramic
P – Plastic
S – Standard 5V ±5%, 0° to 70°C
E – Extended 5V ± 5% -40° to 85°C
M – Military 5V ±10% -55° to 125°C

Example:

AX205556

Z80-DMA CS (Ceramic–Standard range)

10460 Bubb Road, Cupertino, California 95014
Telephone: (408) 446-4666 TWX 910-338-7621

Zilog

03-0013-02